
Resource-Aware Annotation through Active Learning

Dissertation

zur Erlangung des Grades eines

D o k t o r s d e r N a t u r w i s s e n s c h a f t e n

der Technischen Universität Dortmund
am Fachbereich Informatik
von

Katrin Tomanek

Dortmund
2010

Tag der mündlichen Prüfung: 15.04.2010

Dekan: Prof. Dr. Peter Buchholz

Gutachter:

Prof. Dr. Katharina Morik

Prof. Dr. Udo Hahn

Acknowledgements

First of all, I would like to thank my supervisors Prof. Dr. Katharina Morik (University of Dortmund) and Prof. Dr. Udo Hahn (University of Jena). I appreciate Katharina Morik's enthusiasm, her expertise, and her constructive criticism which made me reexamine parts of my work from different perspectives. I am grateful to Udo Hahn for his support in all stages of my thesis by freely sharing his knowledge in numerous helpful discussions, inspiring and motivating me, and giving me the many (and highly appreciated) freedoms I enjoyed during my time as a research assistant in Jena.

Thanks to my (past and present) colleagues at the JULIE Lab in Jena for the various on- and off-topic discussions and many shared coffee and lunch breaks. Thanks also to the student assistants for their valuable support in corpus annotation and software implementation. I would also like to thank all members of the Artificial Intelligence Group in Dortmund – the extensive discussions during the brunch-accompanied colloquia will be kept in good memory.

For proof-reading (parts of) this thesis, my thanks go out to Ingmar Baumgart, Erik Fäßler, Roman Klinger, Florian Laws, Sebastian Merz, and Fredrik Olsson.

Doing a PhD would be only half the fun without all the interesting and insightful discussions with other fellow PhD students. I gained a lot of my understanding of Conditional Random Fields to the extensive but enjoyable discussions with Roman Klinger. Also, my understanding of Active Learning and associated issues has been widened and deepened greatly by discussions with (amongst others) Michael Bloodgood, Ken Dwyer, Robbie Haertel, Florian Laws, Fredrik Olsson, Alexis Palmer, Roi Reichart, Burr Settles, and Andreas Vlachos.

Personally, I am grateful to my parents for the many years they supported my studies and for being there whenever I needed their help. Also my flatmates in Jena must not be forgotten. They generously excused that I hardly ever managed to stick to the cleaning rota and always had food for me when in hard times there was (once again) yawning emptiness in my slot of the fridge. Finally, Sebastian... thanks to your invaluable support, encouragement, and cheering-me-up whenever necessary (which was often enough), this thesis enterprise finally came to a good end. And now, I am happy to plunge into the upcoming adventure with you!

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Resource-Aware Annotation	2
1.3	Outline	6
I	Background	9
2	Supervised Machine Learning	11
2.1	Basic Concepts	11
2.2	Approaches to Classification Learning	14
2.2.1	Probabilistic Models	14
2.2.2	Maximum Margin Classification	23
2.2.3	Decision Tree Induction	25
2.3	Evaluating Models	26
3	Active Learning	29
3.1	Selective Sampling	29
3.2	Approaches to Greedy Active Learning	31
3.2.1	Statistically Optimal Active Learning	32
3.2.2	Expected Model Change	35
3.2.3	Uncertainty Sampling	36
3.2.4	Query-by-Committee	37
3.2.5	Representativeness-aware Approaches	40
3.3	Adaptations for Feasibility	41
3.4	Sampling Complexity Bounds	42
3.5	Related Fields of Research	43
3.6	Applications of Active Learning	44
3.7	Summary	45
II	Towards Resource-Aware Annotation through Active Learning	47

4	Active Learning for Named Entity Recognition	49
4.1	Named Entity Recognition	50
4.1.1	Previous Work	50
4.1.2	Sequence Labeling Problem	51
4.2	Active Learning	52
4.2.1	Related Work	53
4.2.2	NER-specific AL Framework	53
4.2.3	Utility Functions	56
4.3	General Experimental Settings	58
4.3.1	Evaluation Measures	58
4.3.2	Corpora	61
4.3.3	Randomizations	63
4.4	Results	63
4.4.1	Passive Learning	63
4.4.2	Active Learning	64
4.5	Summary and Conclusions	70
5	Monitoring the Active Learning Sampling Process	73
5.1	Related Work	75
5.2	Monitoring and Stopping Committee-based AL	77
5.2.1	Approximating Learning Curves	77
5.2.2	Objectivizing Stopping	80
5.3	Experiments	81
5.3.1	Experimental Settings	81
5.3.2	Results	82
5.4	Summary	90
6	Semi-Supervised Active Learning	93
6.1	Related Work	94
6.2	Combination of Active Learning and Bootstrapping	95
6.2.1	Example Selection	96
6.2.2	Identification of Critical Subsequences	97
6.2.3	Parameters	97
6.3	Experiments and Results	98
6.3.1	Distribution of Confidence Scores	98
6.3.2	Fully-Supervised <i>vs.</i> Semi-Supervised AL	99
6.3.3	Annotated Noun Phrases	100
6.3.4	Selection Characteristics of SESAL	101
6.3.5	Impact of the Confidence Threshold	102
6.3.6	Impact of the Delay Rate	103
6.3.7	Discussion	103

6.4	Summary and Conclusions	104
7	Sample Efficiency in Scenarios of Disparate Selector-Consumer Pairings	107
7.1	Learning Under Sample Selection Bias	108
7.2	Sample Reuse as a Case of Sample Selection Bias	109
7.2.1	Sample Reuse and Reusability	110
7.2.2	Previous Work	112
7.3	Empirical Investigation of Sample Reusability	114
7.3.1	Large-Scale Experiment on Sample Reuse	115
7.3.2	Discussion of Hypotheses	122
7.3.3	Conclusions	133
7.4	Boosting Sample Reusability in the Agnostic Setting	134
7.4.1	Heterogeneous Committees	135
7.4.2	Experiments and Results	135
7.5	Summary and Conclusions	138
III	Active Learning with Multiple Criteria	139
8	Multi-Criteria Active Learning	141
8.1	Decision-Theoretic Formulation of Active Learning	142
8.2	Hierarchical Decision-Making	143
8.3	Multi-Attribute-Value Functions	144
8.4	Previous Work on Multi-Criteria AL	146
8.5	Summary	148
9	Selective Sampling for Multiple Learning Problems	149
9.1	Related Work	150
9.2	Problem Definition	150
9.3	Approaches to Multi-Task Active Learning	151
9.3.1	Alternating Task-Specific Selection	151
9.3.2	Rank Combination	152
9.4	Empirical Assessment	153
9.4.1	Experimental Settings	153
9.4.2	Foreign-Selection	154
9.4.3	Explicit Selection for Multiple Learning Problems	154
9.4.4	Overall Evaluation	157
9.5	Summary and Conclusions	158
10	Reducing Class Imbalance during Active Learning Sampling	159
10.1	Related Work	160

10.2 Approaches	161
10.2.1 Modification of the Selector	162
10.2.2 Post-Processing of the Selection	162
10.2.3 Hierarchical Selection	163
10.2.4 Combined Metric	164
10.3 Experiments	165
10.3.1 Experimental Settings	165
10.3.2 Results	167
10.3.3 Discussion	173
10.4 Summary and Conclusions	174
11 Cost-Sensitive Active Learning	177
11.1 Previous Work	177
11.2 The MUC7 _T Corpus	179
11.2.1 Annotation Procedure	180
11.2.2 Corpus Statistics	182
11.2.3 Annotation Time Analysis	183
11.3 Evaluation of Active Learning with Real Costs	184
11.3.1 Default Active Learning	184
11.3.2 Semi-Supervised AL	186
11.3.3 Conclusions	188
11.4 Cost-Sensitive Active Learning	188
11.4.1 Approaches to Combining Utility and Cost	188
11.4.2 Evaluation	190
11.4.3 Conclusions	195
11.5 Summary	195
IV Active Learning in Practice	197
12 Environment for Active Learning-Driven Annotation	199
12.1 System Architecture	199
12.1.1 Administration Client	200
12.1.2 Annotation Client	202
12.1.3 Annotation Repository	204
12.1.4 Active Learning Component	205
12.2 Large-Scale Annotation of Biomedical Documents	205
12.3 Conclusions	206
13 Survey on the Practical Usage of Active Learning	207
13.1 Survey Set-Up	207

13.2 Questions and Answers	208
13.3 Discussion and Conclusions	211
V Conclusions	213
14 Summary and Perspectives	215
14.1 Summary of Contributions and Achievements	215
14.2 Perspectives	218
A Notation	221
B Abbreviations	223
C Joint Publications	225
D Additional Material for Chapter 7	227
D.1 Correlation Coefficients	227
D.2 WEKA Parameter Settings	228
D.3 Information on Resampling	228
D.4 Additional Data	228

List of Tables

2.1	Contingency table for binary classification scenario	27
4.1	Standard feature set used for CRF-based NER.	52
4.2	Characteristics of the simulation corpora	62
4.3	Passive learning results on NER	64
4.4	Sampling complexity of AL based on different utility functions	65
4.5	Sampling efficiency of AL based on different utility functions	66
4.6	Percentage reduction of sampling complexity through AL	68
4.7	Characteristics of samples obtained by different sampling strategies	71
5.1	Stopping points according to different stopping criteria	87
6.1	Percentage reduction of annotation effort (annotated tokens)	99
6.2	Percentage reduction of annotation effort (annotated noun phrases)	100
6.3	Characteristics corpora constructed with semi-supervised AL	101
6.4	Distribution of self-tagging errors	101
6.5	Maximal model performance with semi-supervised AL	102
7.1	Overview of selected data sets from UCI repository	115
7.2	REU and RAI scores on UCI data sets	119
7.3	REU and RAI scores on the NER scenarios	120
7.4	Relatedness scores for any combination of two learners	126
7.5	Correlation between reusability and similarity of samples	128
7.6	SIM scores of samples selected with different selectors	129
7.7	REU scores on samples of 100 examples	129
7.8	Correlation between reusability and feature ranking	130
7.9	Feature ranking scores of samples selected with different selectors	130
7.10	REU scores of real-world sample reuse scenario	136
9.1	Overview of simulation corpora	153
9.2	RAI scores of different approaches to multi-task AL	157
10.1	Characteristics of the simulation corpora	166
10.2	RAI scores for the different AL strategies	169

10.3	RAI scores for over-sampling during and after AL	172
10.4	RAI scores for the combination of AL-BOOD and over-sampling	174
10.5	RAI scores for AL-BOOD with different boosting factors	175
11.1	Characteristics of the MUC7 _T corpus	182
11.2	CFP scores and percentage reduction of annotation effort	186
12.1	Overview of annotation endeavour in the STEMNET project	206
D.1	REU scores on UCI data sets for samples of 100 examples	229
D.2	SIM scores on UCI data sets for different selectors	230
D.3	Feature ranking scores on UCI data sets for different selectors	231
D.4	Class distribution in original and resampled UCI data sets	232
D.5	REU and RAI scores on resampled UCI data sets	232

List of Figures

1.1	Number of articles in Wikipedia and PubMed	2
1.2	Number of papers on AL as archived in the ACL Anthology	4
2.1	First-order linear-chain Conditional Random Field	20
2.2	Overview of probabilistic models	23
2.3	Maximum Margin classification with SVMs	24
2.4	Decision Tree for simple classification problem	26
3.1	Informativeness and representativeness in a toy scenario	40
3.2	Toy example for finding a linear separator in \mathbb{R}	43
4.1	NER example	50
4.2	Visualization of sampling complexity and sampling efficiency	60
4.3	Learning curves for selected utility functions	67
5.1	Prototypical learning curve subdivided into three stages	74
5.2	Learning and agreement curves for simulation corpora	83
5.3	Learning and agreement curves for simulation corpora	84
5.4	Learning and agreement curves for the real annotation scenarios	86
5.5	Stopping points shown on learning and agreement curves	88
5.6	Intrinsic stopping criterion for stream-based AL	90
6.1	Sample sentence with high overall utility	93
6.2	Distribution of token-level confidence scores	99
6.3	Learning curves for semi-supervised AL	100
6.4	Semi-supervised AL with varied confidence thresholds	102
6.5	Semi-supervised AL with different delay rates	105
7.1	Different scenarios of sample reusability	111
7.2	Demonstration of adversarial effect of sample selection bias	113
7.3	Visualization of the REU measure	117
7.4	Learning curves for MaxEnt consumer on UCI data sets	118
7.5	Learning curves for sample reuse in the NER scenario	121
7.6	Evaluation of AL for NER with F-score and ACC	122

7.7	AL for NER in a token-selection scenario	123
7.8	Distribution of two samples over a clustering.	127
7.9	Sample reuse on on re-sampled data sets	132
7.10	Real-world sample reuse scenario with heterogeneous committees	137
9.1	Learning curves for foreign-selection scenario	155
9.2	Learning curves for the NER and the PARSE consumer	156
10.1	Entity mention statistics	168
10.2	Learning curves of different F-scores	170
10.3	Learning curves for AL-BOOD and over-sampling	173
11.1	Length distribution of sentences and CNPs	183
11.2	Average annotation times per block	184
11.3	Distribution of annotation times over sentences and CNPs	185
11.4	Cost-sensitive evaluation of AL	186
11.5	SeSAL evaluated by number of tokens and annotation time	187
11.6	Testing different parameter settings for CCS and LRK	191
11.7	Relationship between u_{LC}^s and benefit	193
11.8	Testing different parameter settings for BCR	194
11.9	Comparison of CSAL approaches	195
11.10	Evaluation of the cost-sensitive version SeSAL	196
12.1	System architecture of JANE	200
12.2	Administration client	201
12.3	Graphical user interface of the annotation client	202
12.4	MMA2 annotation editor	203

Chapter 1

Introduction

1.1 Motivation

During recent years we have witnessed an explosion of information, emanating to a great extent from electronic mass media and the Internet. Figure 1.1 visualizes the growth of two prototypical sources of information: the number of articles publicly available through Wikipedia¹ has grown over a ten thousand times since 2001; similarly, the number of scientific articles in the biomedical domain as available through PubMed² is also rapidly growing.

Efficient processing of the masses of information available is crucial in order to gain benefit from the growing amount of knowledge. This includes techniques to filter the relevant from the irrelevant and to extract desired information from unstructured text. *Natural Language Processing* (NLP) aims at providing these techniques using computerized linguistic analysis of natural language text.

Many NLP tasks can be described as classification problems, such as the categorization of articles by their genre or the identification of persons' names within a textual document. NLP tasks are often successfully tackled by *machine learning* (ML) methods that can automatically categorize new data. Therefore, training data needs to be provided, from which the predictive *classification model* is learned. A drawback of ML methods is the need for large amounts of training data in order to learn accurate models. Such training data are textual examples to which the true categories of interest, according to the specific task under scrutiny, are added.

The process of creating training data, i.e., adding the task-specific meta-data to the raw examples, is known as *annotation*. Annotation involves human labor and is thus a costly procedure. A human annotator needs to carefully read through the

¹<http://www.wikipedia.org/>

²PubMed (<http://www.ncbi.nlm.nih.gov/pubmed/>) is one of the major bibliographic databases for biomedical articles and comprises more than 19 million citations (as of December 2009).

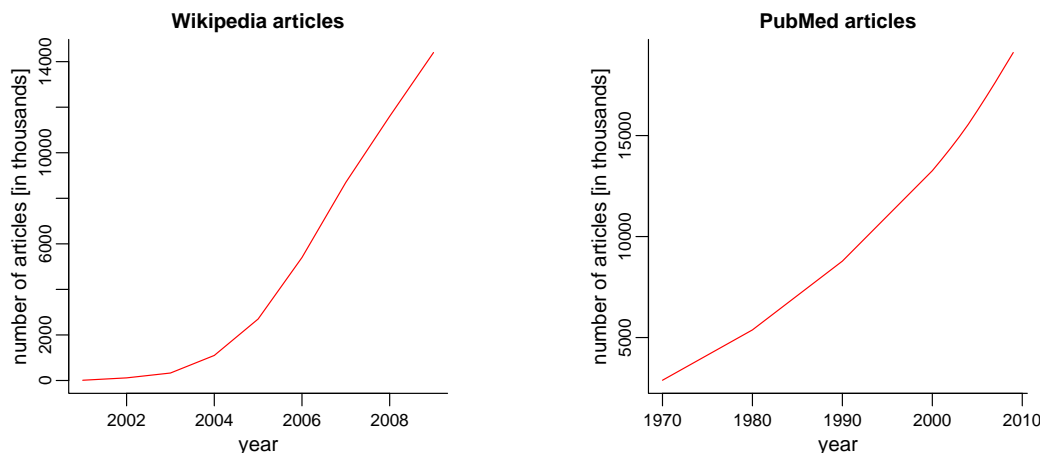


Figure 1.1: Number of articles in Wikipedia and PubMed.

collection of raw text examples and add the meta-data, also called *labels*, where necessary.

Annotation work is monotonous and arduous. Large numbers of examples have to be worked through. Additionally, it is hard to maintain concentration when documents are only sparsely populated with relevant information, which must, however, not be missed. Moreover, annotators often need to be domain experts in order to fully understand the texts themselves and make informed annotation decisions. This is especially the case in the biomedical domain, one of the major application areas for NLP due to the large number of scientific articles published every day (cf. Figure 1.1). The quality of training data, in terms of correctness and completeness of the added labels, has a major impact on the overall performance of the desired information processing solution.

As *human language technology* (HLT) systems based on NLP techniques are increasingly used in both the scientific and industrial contexts, one can foresee a rapidly growing demand for training data. The burden of annotation is already the major bottleneck in the application of established NLP techniques to a variety of concrete scenarios where information needs to be dealt with efficiently.

1.2 Resource-Aware Annotation

For a long time cost awareness has not been the focus of large-scale annotation endeavors. However, with an increasing number of small and medium-sized annotation

projects with limited resources, cost awareness is increasingly being recognized as an important issue. To speed up the creation of training data, several aspects need to be considered, including human-computer interaction through proper user interfaces, rapid development of annotation guidelines, and efficient training of human annotators. Most importantly, however, the costs of creating training data depends on the human interaction needed to annotate the text examples.

In the “traditional” annotation setting, a collection of raw text examples is randomly drawn from a larger pool of available documents. Annotators then work through this collection example by example. The annotated collection may contain lots of redundant information that is not very helpful for learning the model. On the other hand, documents which would very efficiently increase the quality of a model may be missed by random selection.

The annotation effort can be considerably reduced when the examples to be annotated are carefully chosen. In this thesis, a selective sampling method known as *Active Learning* (AL) is comprehensively studied and enhanced to meet the particular requirements of linguistic annotation. From a pool of raw text examples, AL selects exactly those examples that it expects to be most useful for model learning. As a result, the same level of performance can be achieved with much fewer examples than in the standard annotation setting.

AL as an explicit strategy to reduce annotation effort is attracting increasing interest in the NLP community. This is evidenced by the recent workshop on Active Learning for Natural Language Processing³ and by a growing number of papers on AL⁴ as shown in Figure 1.2.

This thesis especially focuses on the appropriateness of AL as a *resource-aware* strategy for annotation of textual documents. From the human resource constraint, the following requirements for AL as an annotation strategy to be practically applicable to real-world annotation endeavours emerge.

Requirement 1 (*Relevant Savings*) *The reduction of annotation effort has to be relevant.*

³This workshop was held in conjunction with the 2009 conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies.

⁴The plot shows the number of papers on AL as archived in the ACL Anthology (<http://aclweb.org/anthology-new/>), a digital archive of research papers in computational linguistics where most of the top conferences in this field host their proceedings. This plot was taken from the slides of a talk held by Burr Settles at the above mentioned workshop (Settles, 2009a).

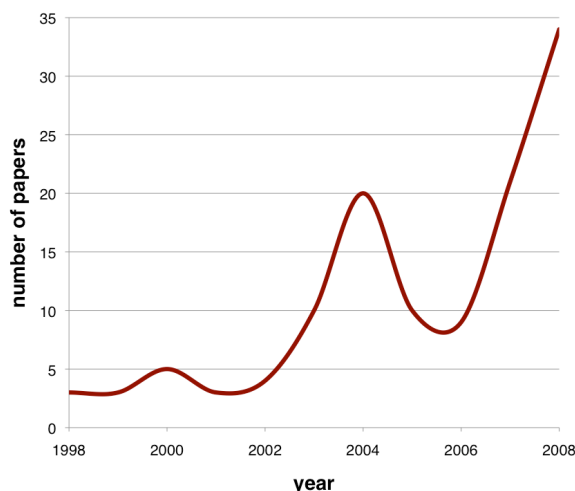


Figure 1.2: Number of papers on AL as archived in the ACL Anthology.

Obviously, only approaches that lead to a practically relevant saving of annotation effort, compared to the standard scenario of random selection of raw examples for annotation, will be considered in a resource-aware annotation strategy.

Requirement 2 (*Rapid Selection*) *Selection of raw examples to be annotated needs to be rapid.*

AL-based annotation is an iterative and interactive process. In every iteration, a small number of raw examples are selected and handed to the human decision maker for annotation. During selection, the annotator is idle. In consequence, approaches to AL which have a near-optimal selection efficiency at the cost of high computational complexity are not desirable.

Requirement 3 (*Generic Approach*) *An AL approach is to be generic and flexible so that application to a broad range of annotation scenarios is possible without major modifications.*

Approaches to AL may be subject to several configuration parameters which have to be set according to the specific learning problem at hand. When AL is applied to a new annotation scenario, such parameters cannot be optimized because in practise, there is no simulation data available. In consequence, the number of configuration parameters required for efficient selection has to be kept low. Moreover, approaches that are extremely variant to such parameters (i.e., where suboptimal parameter

settings easily lead to performance equal to or worse than random selection) are inappropriate.

Requirement 4 (*Monitoring and Stopping*) *Means to monitor the AL-driven annotation process and support for the decision on when to stop the process are essential in order to benefit from selective sampling.*

Without such monitoring facilities, one may miss the point where annotation costs and model performance exhibit the best trade-off. Thus, to cash in the savings achieved by selective sampling, it is necessary to stop early.

Requirement 5 (*Sample Reusability*) *The selected examples must not be overly specific to the model used during annotation, so that the data obtained can also be used to train other models.*

AL selects the examples for manual annotation dependent on the model being used. During annotation time, however, the best model might not be known – model selection can only be performed once training data is available. Hence, the resulting collection of training examples has to be reusable with different models. Otherwise, for each new model to be applied, new training material would be required which is incompatible with the human resource constraints in the annotation scenario.

Requirement 6 (*Multiple Tasks*) *Methods are needed to select data appropriate to multiple tasks.*

Standard AL is designed to select examples for one task only. A task here refers to a specific classification problem for which a model should be learned. Practical HLT systems usually comprise more than one NLP task, so training data for multiple tasks is required. Ideally, annotation for multiple tasks is done in parallel so as to benefit from annotation synergy effects between related tasks. It should thus be possible to select examples efficiently with respect to the multitude of tasks in order to reduce annotation effort further.

Requirement 7 (*Class Imbalance*) *Many data sets exhibit a skewed distribution the classes of interest. This needs to be analyzed and, if necessary, taken into account in the selection of examples during costly data acquisition.*

Standard AL assumes that all classification operations have the same impact on the overall model performance. This constitutes an unrealistic assumption in practise as there are often classes that are less frequent than others, while at the same time their correct recognition may be more valuable. Learning under class imbalance has been intensively studied by the ML community – all solutions, however, assume that labeled data is readily available. In the AL scenario, this is not the case, so that re-balancing of this class imbalance should be done already during AL-based data acquisition.

Requirement 8 (*Incorporation of True Annotation Effort*) *In addition to the usefulness of an example, the human labor needed to annotate a given example also has to be considered during the selection process.*

The overall goal of reducing annotation costs will usually be achieved by minimizing the amount of human labor in the process. Requirements defined above all aimed at reducing the number of examples needed to reach a given level of performance. However, in practise, different examples will vary as to the amount of time required for their annotation. This fact has to be incorporated in the selection process to avoid the selection of useful but overly expensive examples. Additionally, methods must be established to estimate the annotation time needed for a given example.

This thesis analyzes the above mentioned requirements in detail and presents appropriate approaches and modifications to the standard AL framework. Their effectiveness is empirically evaluated in the practically relevant NLP task of *Named Entity Recognition* (NER) which is crucial to most HLT systems. The main objective of this thesis is to provide a comprehensive and widely applicable framework for resource-aware linguistic annotation meeting the above mentioned requirements.

1.3 Outline

This thesis is structured as follows. Part I presents relevant background information necessary to fully understand the thesis. In Chapter 2, we formalize and describe the problem of classification learning in the context of supervised ML. This chapter gives a detailed overview of the models used throughout the thesis and briefly discusses methods to assess the performance of a learned model. Chapter 3 gives a formal definition of AL, followed by a description of several approaches to AL as well as utility functions. After a brief discussion of theoretical sampling complexity bounds of AL, an overview of common fields of AL application is provided.

In Part II, a framework for resource-aware linguistic annotation is presented. In Chapter 4, we begin with a description of the NER task that is used in this thesis as a sample scenario to test the proposed approaches to AL. Addressing Requirements 1, 2, and 3, this chapter describes the basic framework of AL for linguistic annotation using the example of NER. This description is followed by a comprehensive evaluation of its sampling complexity and efficiency. Chapter 5 addresses the question of how the progress of AL can be monitored with the ultimate goal of finding a good stopping point for AL-driven annotation, according to individual cost-benefit trade-offs (Requirement 4). We present a method for approximating the progression of the learning curve and an intrinsic stopping criterion which does not require any parameters to be specified. A novel approach to semi-supervised AL is presented in Chapter 6, addressing Requirement 1. It reduces the actual human annotation effort still further by precisely pointing the annotators towards annotation-relevant regions within the raw text examples. Chapter 7 discusses how samples obtained by AL for a specific model can be reused by other models (Requirement 5). Sample reusability is empirically studied in the context of several classification problems, with the aim of identifying factors that positively or negatively influence reusability. Finally, this chapter presents a novel method for increasing sample reusability for a setting where the final model is not known at selection time.

Part III begins with a formalization of multi-criteria AL given in Chapter 8, an extension of the standard AL framework allowing the incorporation of several criteria into the selection process. Additionally, several approaches are proposed for addressing selection with respect to multiple criteria. The subsequent three chapters describe concrete instantiations of multi-criteria AL applying the proposed methods. Chapter 9 presents an application of multi-criteria AL to the problem of selecting examples appropriate to learning several tasks simultaneously as specified by Requirement 6. The approaches are proposed and evaluated for the combination of NER and syntactic parsing. With respect to Requirement 7, approaches to tackling class imbalance upfront during AL-based data acquisition are presented and evaluated in Chapter 10. Incorporation of true annotation effort in the AL selection process (Requirement 8) is considered in Chapter 11. We first evaluate our cost-insensitive approaches to AL considering true annotation time, and then present and compare methods for incorporating annotation time as a secondary criterion in the AL selection process.

Finally, in Part IV we report on the practical application of AL for the annotation of linguistic data. Chapter 12 describes our implementation of an integrated annotation environment allowing for AL-driven annotation. This environment has been comprehensively used to create training data for NER in biomedical and scientific documents. Secondly, Chapter 13 reports on a survey conducted to discover the actual usage of AL as an annotation strategy within the NLP community. The sur-

vey outlines critical issues that need to be addressed in order to establish AL as a widely-used annotation strategy.

Lastly, Chapter 14 summarizes the key contributions and achievements of this work with respect to the above-mentioned requirements. The thesis ends with a discussion of problems that are still outstanding and future work necessary to consolidate AL as a resource-aware annotation strategy.

Part I

Background

Chapter 2

Supervised Machine Learning

This chapter first sketches basic principles and concepts of *supervised* machine learning with a focus on *classification* learning. This is followed by a description of common approaches to classification learning which were applied in this thesis. Finally, methods to evaluate the performance of the learning results are presented.

2.1 Basic Concepts

The goal of machine learning (ML) is to find a function $g : \mathcal{X} \rightarrow \mathcal{Y}$ which maps an observation $x \in \mathcal{X}$ to its target value $y \in \mathcal{Y}$.

Definition 1 (*Observation, Target*) An observation x is an object or event observed in the data. \mathcal{X} is the set of all possible observations. The target $y \in \mathcal{Y}$ is the outcome of a mapping and \mathcal{Y} is the set of all possible target values.

Two types of learning problems can be distinguished depending on the target space \mathcal{Y} : *regression learning* assumes a continuous target space $\mathcal{Y} = \mathbb{R}$, *classification learning*, in contrast, constitutes a mapping into a discrete and finite target set $\mathcal{Y} = \{y_1, \dots, y_c\}$. The focus of this thesis is on classification learning, only. The target values $y \in \mathcal{Y}$ in classification learning are called *classes*. Binary classification means that $|\mathcal{Y}| = 2$, the classes are known as *positive* and *negative* class, and the target set is then given by $\mathcal{Y} = \{-1, +1\}$. In *multi-class* classification, $|\mathcal{Y}| > 2$.

When applied to NLP tasks, the classes of interest are, for example, semantic types. In a document classification scenario, a text document constitutes an observation x . The classes could be the language a document is written in so that $\mathcal{Y} = \{\text{en}, \text{de}, \text{es}, \text{fr}\}$.

Depending on the respective ML approach used, different representations of the target function g are chosen. The actual representation of g models the particular learning problem at hand.

Definition 2 (Model) A model is a mathematical equation describing the relationship between a dependent variable y and one or more independent variables x given a specific target function representation and a set of free parameters θ .

A very simple model describing a linear relationship between observation x and target value y is given by $y = \lambda x$ where the free parameter λ has to be fitted according to the training data. An important step in designing a ML approach for a specific problem is the choice of an appropriate model (Mitchell, 1997). In the next section, several models commonly applied to problems of NLP are discussed.

While “model” refers to a class of target function representations, the term “model” is in literature often used to refer to the *fitted* model, i.e., the combination of the target function representation and the specific parameter set $\hat{\theta}$. We use the term “model” in the same manner when the meaning is clear and unambiguous.

Definition 3 (Learning Algorithm) A learning algorithm $T(\mathcal{L})$ is applied to estimate the model parameters $\theta = (\lambda_1, \dots, \lambda_k)$ from the training data \mathcal{L} . Parameter estimation constitutes the actual learning step.

There are several learning paradigms including supervised learning as one of the most important ones. Under the *supervised learning* paradigm, the parameters θ are estimated with respect to the training set \mathcal{L} containing so-called *labeled* examples, i.e., tuples (x, y) , so that $\mathcal{L} = \{(x_i, y_i)\}_{i=1}^m$. Learning here means searching through the parameter space $\Theta \subset \mathbb{R}^k$ to find the parameter set θ^* which best fits the knowledge given by \mathcal{L} . Learning algorithms often make use of optimization techniques such as iterative-scaling methods (Darroch and Ratcliff, 1972) to solve the search problem efficiently.

The terms “classifier” and “model” are often used synonymously. However, a classifier is constructed from a model and makes the final classification decision. Probabilistic models, for example, model a probability distribution $P(x|y)$ and the classifier decides which probability value leads to which final target class $y \in \mathcal{Y}$.

Definition 4 (Classifier) A classifier combines a model θ with a decision rule and is given by a function $g_\theta(x) = \hat{y}$ which returns the predicted target class \hat{y} for a new observation x .

For the sake of simplicity we have so far assumed that the model operates on the observation x directly. Instead, an observation is usually characterized by a vector of *features*.

Definition 5 (Feature Representation) An observation x is characterized by a feature vector $\vec{x} \in \mathbb{R}^k$ which constitutes an attribute-value representation of x and is generated by a feature generating function $F : \mathcal{X} \rightarrow \mathbb{R}^k$. F is composed by a set of feature functions $f : \mathcal{X} \rightarrow \mathbb{R}$ so that $\vec{x} = (f_1(x), \dots, f_k(x))$.

Models are usually formulated based on the feature vector representation \vec{x} of an observation x and for each element of \vec{x} , a parameter λ_k is used in the model. In the following, we use the term “feature” to refer to a particular feature function f_j .

As an example, in the document classification scenario the feature $f_j(x)$

$$f_j(x) = \begin{cases} 1 & \text{if } x \text{ contains word "bonjour"} \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

indicates whether the word “bonjour” is contained in the observed document x .

Many NLP tasks can be formulated as learning problems and solved successfully by classification learning. This includes, as a pioneer task, document classification but also much more complex tasks. In many situations, separate classification of isolated observations $x \in \mathcal{X}$ constitutes problematic assumptions on the (statistical) independence amongst the individual observations. This is, for example, the case for words in a sentence where each word should be mapped to its associated part of speech it has in the respective sentence.¹ Words in natural language texts are not an arbitrary accumulation. Instead, grammatical and semantic constraints hold and imply dependencies between the words.

Data subject to such dependencies is called relational. The goal of *relational learning* is to appropriately handle such data by explicitly modeling dependencies between the data points (Getoor and Taskar, 2007). We here consider relational learning as a variant of classification learning where the observation and the target space are of a higher dimension. Relation learning is about learning a function $g : \mathcal{X}^n \rightarrow \mathcal{Y}^n$ which maps a vector of observations $\vec{x} \in \mathcal{X}^n$ to a vector of target values $\vec{y} \in \mathcal{Y}^n$. The elements in $\vec{x} = (x_1, \dots, x_n)$ and $\vec{y} = (y_1, \dots, y_n)$ are subject to a specific order or structure. We here consider only the case of linear sequences. When applied to NLP tasks, such a sequence may be the words in their original order in a piece of natural language text, such as a phrase or a sentence.

Analogously to Definition 5, a single element x_i of an observation \vec{x} is characterized by a feature vector \vec{x}_i obtained by the feature generating function $F(x_i)$ which is composed of several feature functions $f_j(\vec{x}, i) = x_i$ with $x_i \in \mathbb{R}$ so that $\vec{x}_i =$

¹The part of speech of a word describes how a word is used in, e.g., a sentence. Common parts of speech are “nouns”, “adjectives”, or “verbs”, amongst others.

$(f_1(\vec{x}, i), \dots, f_k(\vec{x}, i))$ and $\vec{x} = (F(x_1), \dots, F(x_n))$. In a specific feature function $f_j(\vec{x}, i)$, $1 \leq i \leq n$ specifies the position in the observation sequence \vec{x} .

As an example, a feature function on observation $\vec{x} = (\text{We}, \text{walk}, \text{home})$ could be

$$f_j(\vec{x}, i) = \begin{cases} 1 & \text{if } x_i = \text{"walk"} \text{ and } (x_{i-1} = \text{"we"} \text{ or } x_{i-1} = \text{"I"}) \\ 0 & \text{otherwise.} \end{cases} \quad (2.2)$$

2.2 Approaches to Classification Learning

As briefly stated in the previous section, depending on the specific approach to classification learning, the problem to be learned may be modelled in different forms. Amongst the various models described in the literature on ML, the most common approaches used for NLP tasks include the representation of the target function by probabilistic models, maximum margin models, and decision trees. In the following, several models which fall into these three classes are described. Besides the model itself, the standard approaches to find good parameter estimates are described.

Since Conditional Random Fields are used as the main model of choice throughout the complete thesis, a more detailed description is given. In addition, Maximum Entropy models, on which Conditional Random Fields fundamentally build, are also described in greater detail. As for the other models, only a short introduction describing the approach in very general.

2.2.1 Probabilistic Models

A probabilistic model describes the data by a conditional probability distribution $P(y|x)$. The parametric form of $P_\theta(y|x)$ depends on the respective modelling paradigm chosen; the parameters $\theta = (\lambda_1, \dots, \lambda_k)$ of the probability distribution are estimated by statistical inference. We assume all classifiers $g_\theta(x)$ based on probabilistic models to apply the Bayesian classification rule so that

$$g_\theta(x) = \operatorname{argmax}_{y' \in \mathcal{Y}} P_\theta(y'|x) \quad (2.3)$$

which means that the target class $y^* = g_\theta(x)$ is the class with the highest probability given a specific observation x . The following description of probabilistic models is largely based on a technical report by Klinger and Tomanek (2007). Details omitted in this section can be found in this report.

2.2.1.1 Naïve Bayes Model

According to Bayes' Law, the probability distribution $P(y|x)$ can be written as

$$P(y|x) = \frac{P(y)P(x|y)}{P(x)}. \quad (2.4)$$

In a classifier based on the Bayesian classification rule (Equation 2.3), the denominator $P(x)$ is a constant and so not needed for classification. The numerator can be written as a joint probability

$$P(y)P(x|y) = P(y, x). \quad (2.5)$$

When the above probabilities are formulated on the feature representation \vec{x} of x , direct computation of $P(y)P(\vec{x}|y) = P(y, \vec{x})$ is complex, especially when the number of elements in \vec{x} is high. Through the application of the chain rule

$$\begin{aligned} P(\vec{x}) &= P(x_1, \dots, x_k) \\ &= \prod_{j=2}^k P(x_j | x_{j-1}, \dots, x_1) \end{aligned} \quad (2.6)$$

a general decomposition of the joint probability can be formulated as

$$P(y, \vec{x}) = P(y) \prod_{j=2}^k P(x_j | x_{j-1}, \dots, x_1). \quad (2.7)$$

Under the assumption of conditional independence of all features, which is usually done for feasibility, $P(x_j | y, x_i) = P(x_j | y)$ holds for all $i \neq j$. Based on this simplification known as the *Naïve Bayes assumption*, the *Naïve Bayes* (NB) model is formulated as²

$$P(y|\vec{x}) \propto P(y, \vec{x}) = P(y) \prod_{j=1}^k P(x_j | y). \quad (2.8)$$

This probability distribution is less complex than the one formulated in Equation 2.7. Dependencies between the features are explicitly ignored often leading to an imperfect representation of the data. Nevertheless, the NB model performs surprisingly well in many real-world applications.

² $A \propto B$ indicates that A is proportional to B. Here, proportionality is given because of the omission of the denominator.

2.2.1.2 Hidden Markov Models

To predict a sequence of class variables $\vec{y} = (y_1, \dots, y_n)$ for an observation sequence $\vec{x} = (x_1, \dots, x_n)$, a simple sequence model can be formulated as a product over individual NB models where each individual model describes the “local” relationship between the i -th element of \vec{x} and the i -th element of \vec{y} :

$$P(\vec{y}, \vec{x}) = \prod_{i=1}^n P(y_i) \cdot P(x_i|y_i). \quad (2.9)$$

In this simple model, dependencies between individual sequence positions are not taken into account and each observation x_i depends only on the class variable y_i at the respective sequence position. No transition probabilities are taken into account. To more appropriately model sequences of observations, state transition probabilities are added to the model so that we obtain the following joint probability

$$P(\vec{y}, \vec{x}) = \prod_{i=1}^n P(y_i|y_{i-1})P(x_i|y_i). \quad (2.10)$$

which leads to the well-known *Hidden Markov Model* (HMM).

In this model, each target state y_i is assumed to depend only on its immediate predecessor and each observation x_i only on the target state y_i . While this claim is made for feasibility reasons, a shortcoming is the assumption of conditional independence between the elements in \vec{x} . Conditional Random Fields address exactly this problem.

The three basic problems in the context of HMMs are (a) to estimate the probability for a predicted label sequence \vec{x} , (b) inference, and (c) finding the best label sequence. The Viterbi and the Forward-Backward algorithm (Rabiner, 1989) have been proposed to solve these problems efficiently. We refrain from a detailed description at this point but will come back to these algorithms in the context of Conditional Random Fields below.

2.2.1.3 Multinomial Logit Model

The multinomial logit model, known as *Maximum Entropy* (MaxEnt) model in the NLP community, is a generalization of logistic regression to allow an arbitrary number of classes. We describe this class of models in greater detail because Conditional Random Fields fundamentally build on them. The following explanations are based on the MaxEnt tutorial of Berger et al. (1996).

Parameter estimation of the MaxEnt model is based on the *Principle of Maximum Entropy* which states that if incomplete information about a probability distribution is available, the only unbiased assumption that can be made is a distribution which is as uniform as possible given the available information (Jaynes, 1957). Thus, the best probability distribution is the one which maximizes the entropy given the constraints from the training material.³ For $P_\theta(y|x)$, the conditional entropy

$$H_\theta(y|x) = - \sum_{(x',y') \in \mathcal{X} \times \mathcal{Y}} P_\theta(y', x') \log P_\theta(y'|x') \quad (2.11)$$

is applied. The goal is now to find a conditional probability distribution $P_\theta(y|x)$ which maximizes $H(y|x)$ and is consistent with the training set \mathcal{L} . This leads to the following target function, later referred to as *primal problem*

$$P_\theta(y|x) = \operatorname{argmax}_{P \in \mathcal{W}} H_\theta(y|x) \quad (2.12)$$

where \mathcal{W} denotes the set of all conditional probability distributions $P(x|y)$ consistent with the training set \mathcal{L} . What is meant by “consistent” is explained in detail below.

The binary-valued feature functions $f_j(x, y)$ are here defined to depend on both the observation x and the class variable y . An example for such a function in context of the document classification scenario is

$$f_j(x, y) = \begin{cases} 1 & \text{if } y = \text{verb and } x = \text{walk} \\ 0 & \text{otherwise} \end{cases} \quad (2.13)$$

The empirical expectation of each feature function $f_j(x, y)$ can be obtained by simply counting the occurrences of $f_j(x, y)$ in \mathcal{L} :

$$\tilde{E}(f_j) = \frac{1}{|\mathcal{L}|} \sum_{(x',y') \in \mathcal{L}} f_j(x', y'). \quad (2.14)$$

Analogously, the expected value of a feature given the model distribution is

$$E(f_j) = \frac{1}{|\mathcal{L}|} \sum_{x' \in \mathcal{L}} \sum_{y' \in \mathcal{Y}} P_\theta(y'|x') f_j(x', y'). \quad (2.15)$$

Only observations in the training data $x \in \mathcal{L}$ are considered while all possible target values $y \in \mathcal{Y}$ are taken into account. In many applications the set \mathcal{Y} contains only a small number of elements so that the summation over all $y \in \mathcal{Y}$ does not constitute a computational problem and $E(f_j)$ can be calculated efficiently.

³The entropy of a random variable $X = \{x_1, \dots, x_n\}$ is given by $H(X) = - \sum_{i=1}^n P(x_i) \log P(x_i)$.

Equation 2.12 postulates that the distribution $P_\theta(y|x)$ be consistent with the evidence found in the training material. That means, for each f_j its expected value on the empirical distribution must be equal to its expected value on the particular model distribution leading to the following k constraints

$$\tilde{E}(f_j) = E(f_j). \quad (2.16)$$

As an additional constraint, a proper conditional probability is required so that

$$P_\theta(y|x) \geq 0 \text{ for all } x, y \quad \sum_{y' \in \mathcal{Y}} P_\theta(y'|x) = 1 \text{ for all } x. \quad (2.17)$$

Finding a probability distribution $P_\theta(y|x)$ which conforms to these constraints can be formulated as a constrained optimization problem. For each constraint a Lagrange multiplier is introduced. This leads to the following function with $\theta' = (\theta, \lambda_{k+1})$:

$$\begin{aligned} \ell(\mathcal{L}, \theta') = & \underbrace{H_\theta(y|x)}_{\substack{\text{primal problem} \\ \text{equation 2.12}}} + \sum_{j=1}^k \lambda_j \underbrace{\left(E(f_j) - \tilde{E}(f_j) \right)}_{\substack{\stackrel{!}{=} 0 \\ \text{constraints from} \\ \text{equation 2.16}}} + \lambda_{k+1} \underbrace{\left(\sum_{y' \in \mathcal{Y}} P_\theta(y'|x) - 1 \right)}_{\substack{\stackrel{!}{=} 0 \\ \text{constraint from} \\ \text{equation 2.17}}} \end{aligned} \quad (2.18)$$

Maximizing this equation we obtain the general formulation of a MaxEnt model

$$P_\theta(y|x) = \frac{1}{Z_\theta(x)} \exp \left(\sum_{j=1}^k \lambda_j f_j(x, y) \right) \quad (2.19)$$

with the normalization factor

$$Z_\theta(x) = \sum_{y' \in \mathcal{Y}} \exp \left(\sum_{j=1}^k \lambda_j f_j(x, y') \right) \quad (2.20)$$

which guarantees the constraint from Equation 2.17. Parameters for a MaxEnt model are optimized by means of maximization of the log-likelihood. Details are skipped here because the log-likelihood maximization is described below in more detail for Conditional Random Fields.

2.2.1.4 Conditional Random Fields

Conditional Random Fields (Lafferty et al., 2001) are probabilistic models for relational data $\vec{x} \in \mathcal{X}^n$ and $\vec{y} \in \mathcal{Y}^n$. A general formulation of *Conditional Random Fields* (CRFs) is that of an undirected graphical model. A probabilistic graphical model is the diagrammatic representation of the probability distribution where nodes represent random variables and edges define dependencies between these. An advantage of graphical modeling is that it allows for a decomposition, or factorization, of a probability distribution making complex computations, such as inference, much more efficient (Bishop, 2006).

A general CRF is defined as

$$P_{\theta}(\vec{y}|\vec{x}) = \frac{1}{Z_{\theta}(\vec{x})} \prod_{C' \in \mathcal{C}} \Psi_{C'}(\vec{x}_{C'}, \vec{y}_{C'}) \quad (2.21)$$

$$Z_{\theta}(\vec{x}) = \sum_{\vec{y}' \in \mathcal{Y}} \prod_{C' \in \mathcal{C}} \Psi_{C'}(\vec{x}_{C'}, \vec{y}') \quad (2.22)$$

where $\Psi_C \geq 0$ are individual, conditionally independent factors of the model, also so-called *potential-functions*. Factorization is performed in such a way that conditionally independent nodes do not appear within the same factor, i.e., they belong to different *cliques* $C' \in \mathcal{C}$.

In general, CRFs can model arbitrary underlying graph structures. In this thesis, we consider only a special form of CRFs known as *first-order linear-chain* CRFs which are structured as a linear chain where the target variables are modeled as a sequence $\vec{y} = (y_1, \dots, y_n)$. In accordance with the general notation of graphical models, such a model is given by

$$P_{\theta}(\vec{y}|\vec{x}) = \frac{1}{Z_{\theta}(\vec{x})} \prod_{i=1}^n \Psi_i(\vec{x}, \vec{y}), \quad (2.23)$$

with factors $\Psi_i(\vec{x}, \vec{y}) = \exp\left(\sum_{j=1}^k \lambda_j f_j(y_{i-1}, y_i, \vec{x}, i)\right)$ and n as the length of the observation and the target sequence. The first-order characteristic is given by the definition of Ψ_i on y_{i-1} and y_i . Figure 2.1 shows a graphical representation of this model. We can now write a linear-chain CRF as

$$P_{\theta}(\vec{y}|\vec{x}) = \frac{1}{Z_{\theta}(\vec{x})} \cdot \prod_{i=1}^n \exp\left(\sum_{j=1}^k \lambda_j f_j(y_{i-1}, y_i, \vec{x}, i)\right). \quad (2.24)$$

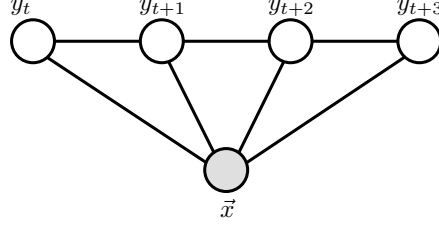


Figure 2.1: Graphical representation of a first-order linear-chain CRF by an independency graph. Any two nodes y_t and y_v with $1 \leq t \leq v-2 \leq n$ are conditionally independent.

This formulation shows the close relationship of CRFs to the MaxEnt model given in Equation 2.19. The normalization coefficient is given by

$$Z_{\theta}(\vec{x}) = \sum_{\vec{y}' \in \mathcal{Y}} \exp \left(\sum_{i=1}^n \sum_{j=1}^k \lambda_j f_j(y'_{i-1}, y'_i, \vec{x}, i) \right). \quad (2.25)$$

Training As in the MaxEnt model, parameters $\theta = (\lambda_1, \dots, \lambda_k)$ are set to maximize the log-likelihood function on the training data $\mathcal{L} = \{(\vec{x}_i, \vec{y}_i)\}_{i=1}^m$:

$$\ell(\mathcal{L}, \theta) = \sum_{(\vec{x}', \vec{y}') \in \mathcal{L}} \log P_{\theta}(\vec{y}' | \vec{x}') - \sum_{j=1}^k \frac{\lambda_j^2}{2\sigma^2} \quad (2.26)$$

Penalization by the term $\sum_{j=1}^k \frac{\lambda_j^2}{2\sigma^2}$ is done to avoid overfitting: σ^2 models the trade-off between fitting exactly the observed feature frequencies and the squared norm of the weight vector. The smaller the values, the smaller the weights are forced to be, so that the chance that few high weights dominate is reduced.

After reformulation, the partial derivations of $\ell(\mathcal{L}, \theta)$ are given by⁴

$$\frac{\partial \ell(\mathcal{L}, \theta)}{\partial \lambda_j} = \tilde{E}(f_j) - E(f_j) - \frac{\lambda_j}{\sigma^2} \quad (2.27)$$

where $\tilde{E}(f_j)$ is the empirical expectation of feature f_j . $E(f_j)$ is the model expectation of f_j and can be written as

$$E(f_j) = \sum_{\vec{x}' \in \mathcal{L}} \sum_{\vec{y}'' \in \mathcal{Y}^n} P_{\theta}(\vec{y}'' | \vec{x}') \cdot \sum_{i=1}^n f_j(y''_{i-1}, y''_i, \vec{x}', i) \quad (2.28)$$

⁴We elaborate in detail on the steps omitted here in (Klinger and Tomanek, 2007).

This is in close relation to the definition of these expectations for the MaxEnt model (Equations 2.14 and 2.15). While $\tilde{E}(f_j)$ can easily be calculated by just counting the occurrences of f_j in \mathcal{L} , direct computation of $E(f_j)$ is infeasible due to the sum over all possible label sequences $\vec{y}'' \in \mathcal{Y}^n$. This is a major difference compared to MaxEnt models which exhibit a considerably lower training complexity as there the sum is over individual labels instead of label sequences.

A dynamic programming approach known as the Forward-Backward algorithm (Rabiner, 1989) can be applied to solve this problem efficiently for linear-chain CRFs. The Forward-Backward algorithm has a run-time of $\mathcal{O}(|\mathcal{Y}|^2 n)$, so it is linear in the length of the sequence and quadratic in the number of labels. Forward (α) and backward (β) scores are defined by

$$\begin{aligned}\alpha_i(y|\vec{x}) &= \sum_{y' \in T_i^{-1}(y)} \alpha_{i-1}(y'|\vec{x}) \cdot \Psi_i(\vec{x}, y', y) \\ \beta_i(y|\vec{x}) &= \sum_{y' \in T_i(y)} \beta_{i+1}(y'|\vec{x}) \cdot \Psi_i(\vec{x}, y, y')\end{aligned}$$

where $\Psi_i(\vec{x}, a, b) = \exp\left(\sum_{j=1}^k \lambda_j f_j(a, b, \vec{x}, i)\right)$, $T_i(y)$ is the set of all successors of a state y at a specified position i , and $T_i^{-1}(y)$ is the set of predecessors. Normalized forward and backward scores are inserted into Equation 2.28 replacing $\sum_{\vec{y}'' \in \mathcal{Y}^n} P_\theta(\vec{y}''|\vec{x}')$. After this conversion, $\ell(\mathcal{L}, \theta)$ can be efficiently estimated by algorithms such as iterative-scaling (Darroch and Ratcliff, 1972; Huang et al., 2009).

Inference and Probabilities For a specific observation sequence \vec{x} , we define the best label sequence \vec{y}^* to be the one which maximizes the a-posteriori probability. For this optimization, the denominator of Equation 2.24 can be skipped as it is a constant, so that the CRF sequence classifier is given by

$$g_\theta(\vec{x}) = \operatorname{argmax}_{\vec{y}' \in \mathcal{Y}^n} \exp\left(\sum_{i=1}^n \sum_{j=1}^k \lambda_j f_j(y'_{i-1}, y'_i, \vec{x}, i)\right). \quad (2.29)$$

The Viterbi algorithm (Rabiner, 1989) is an efficient solution to the problem of finding the best sequence $\vec{y}^* = g(\vec{x})$, called *Viterbi sequence* henceforth.

As will be discussed in the next chapter, many approaches to active learning are based on the model's confidence in its predicted target value, i.e., the a-posteriori probability $P_\theta(y|x)$. To calculate the conditional probability $P_\theta(\vec{y}|\vec{x})$ for a CRF, the normalization factor needs to be calculated. Once again, naïve calculation of Z_θ

is infeasible because of the summation over *all possible* label sequences (cf. Equation 2.25). Applying the Forward-Backward algorithm on \vec{x} we retrieve the respective forward and backward scores and the normalization factor can be efficiently calculated as the sum over all recursively defined forward scores

$$Z_{\theta}(\vec{x}) = \sum_{y' \in \mathcal{Y}} \alpha_n(y|\vec{x}). \quad (2.30)$$

so that we can efficiently calculate a normalized probability score for a label sequence as defined in Equation 2.24.

Instead of a probability for the complete sequence, one can also obtain different forms of marginal probabilities such as, e.g., the marginal probability of label $y' \in \mathcal{Y}$ at position i given an input sequence \vec{x} . This is done using forward and backward scores so that

$$P_{\theta}(y_i = y'|\vec{x}) = \frac{\alpha_i(y'|\vec{x}) \cdot \beta_i(y'|\vec{x})}{Z_{\theta}(\vec{x})}. \quad (2.31)$$

The three basic problems in the context of linear-chain CRFs, being (a) estimation of model parameters $\theta = (\lambda_1, \dots, \lambda_k)$, (b) finding the best label sequence \vec{y}^* , and (c) estimating the probability of a label sequence, can be solved by the algorithms developed for HMMs after slight modification of forward and backward scores. In the case of a non-linear structure, training and inference on CRFs is much more complex. In this thesis, only CRFs with a linear-sequence structure are considered.

2.2.1.5 From Naïve Bayes to Conditional Random Fields

The four models discussed in this section make different independence assumptions over the features of an observation. While simplifying independence assumptions are made for the sake of feasibility for NB models and HMMs, MaxEnt and CRFs refrain from making assumptions on the dependency of the elements of \vec{x} .

The relationship between the four probabilistic models discussed in this section is based on the type of probability distribution they model and whether they are based in relational, or sequence, data or not (see Figure 2.2). A HMM can be understood as the sequence version of a NB model: Instead of single independent decisions, a HMM models a linear sequence of decisions. Accordingly, CRFs can be understood as the sequence version of MaxEnt models. NB and HMM are generative models because they model the distribution of observations and targets. In contrast, MaxEnt models and CRFs are discriminative models which only model the conditional distribution $P(y|x)$ and refrain from modelling $P(x)$. MaxEnt models and CRFs can handle large

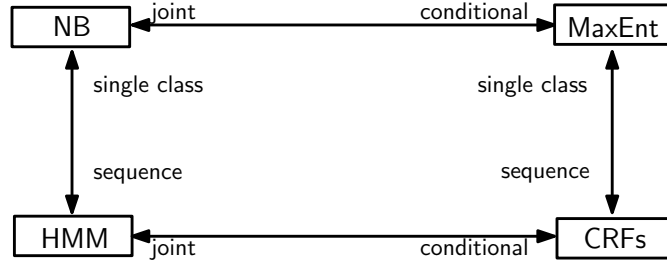


Figure 2.2: Relationship between probabilistic models discussed in this section. Depicted aspects are joint versus conditional probability, single class prediction versus sequence classification.

sets of, presumably, overlapping features. Due to the independence assumptions made by NB models, such feature sets would lead to poor classifier accuracy.

2.2.2 Maximum Margin Classification

In contrast to probabilistic methods, approaches to maximum margin classification are based on models which are essentially specified by a separating hyperplane in the multi-dimensional input space \mathbb{R}^k given by the feature representation \vec{x} of the observation x . The best separating hyperplane is the one which has the maximum distance – called margin – to all points $(x, y) \in \mathcal{L}$. In the following, we will focus on Support Vector Machines (SVMs), which are probably the most prominent approach to maximum margin classification.

For SVMs, the target space consists of two classes with $\mathcal{Y} = \{-1, +1\}$. A hyperplane can completely be described by a vector \vec{w} in \mathbb{R}^k defining the orientation of the hyperplane in the input space and an offset $b \in \mathbb{R}$. The hyperplane is defined as

$$\langle \vec{w}, \vec{x} \rangle + b = 0. \quad (2.32)$$

The vector \vec{w} is a weight vector and is perpendicular to the hyperplane. Given the data is *linearly separable*, such a hyperplane must exist. The margin is defined as the perpendicular distance of the closest point(s) to the hyperplane. These points are called *support vectors*. Figure 2.3 shows such a separating hyperplane, some data points, and illustrates the meaning of \vec{w} and b .

The distance of an observation x , represented by a feature value vector \vec{x} , to the

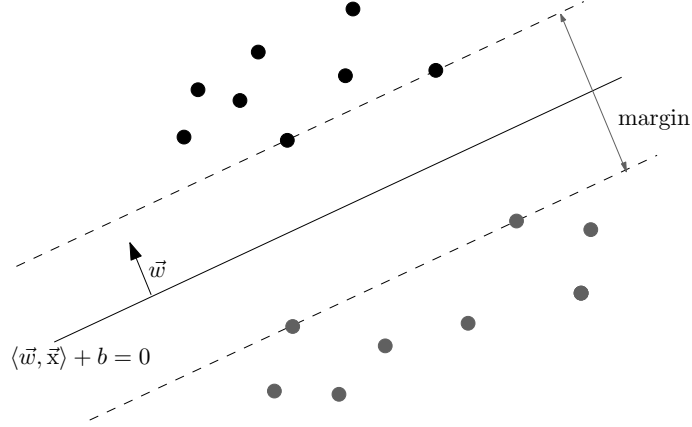


Figure 2.3: SVM for a simple classification scenario: The solid line represents the separating hyperplane. Points on the dashed lines are the support vectors.

hyperplane is given by

$$\begin{aligned} d(\vec{x}) &= \langle \vec{w}, \vec{x} \rangle + b \\ &= \sum_{(x_i, y_i) \in \mathcal{L}} \lambda_i y_i \langle \vec{x}_i, \vec{x} \rangle + b. \end{aligned} \quad (2.33)$$

Only for the support vectors $\lambda_i \neq 0$. The SVM classifier is given by

$$g_{\vec{w}, b}(x) = \text{sgn}(\langle \vec{w}, \vec{x} \rangle + b) \quad (2.34)$$

so that points x for which $\langle \vec{w}, \vec{x} \rangle + b > 0$ are classified as positive instances, or negative otherwise. This classification approach is one reason why the target classes are constraint to -1 and $+1$.

The distance $d(\vec{x})$ can be interpreted as a measure of confidence of the SVM model in its target prediction. The confidence depends on the distance, the farther points are away from the margin, the more confident is the SVM classifier.

During training, \vec{w} and b are set so that the resulting hyperplane separates the data as well as possible and achieves the highest margin. The correct separation of all training examples is given if $y_i(\langle \vec{w}, \vec{x}_i \rangle + b) \geq 0$ for all $(x_i, y_i) \in \mathcal{L}$. When \vec{w} and b are normalized so that the support vectors satisfy $|\langle \vec{w}, \vec{x}_i \rangle + b| = 1$, then $y_i(\langle \vec{w}, \vec{x}_i \rangle + b) \geq 1$ for all $(x_i, y_i) \in \mathcal{L}$ in case of perfect separation.

To find the best margin, the following constrained optimization problem

$$\ell(\vec{w}, b, \theta) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^n \lambda_i (y_i (\langle \vec{w}, \vec{x}_i \rangle + b) - 1) \quad (2.35)$$

has to be solved. This can be done using standard quadratic programming techniques. For more details see Schölkopf and Smola (2002).

Although SVMs are linear models, they can deal with non-linearly separable data by mapping the feature representation \vec{x} of observations x by a non-linear mapping function into a higher-dimensional feature space \mathcal{H} given a function $\Phi : \mathbb{R}^k \rightarrow \mathcal{H}$. This is done using kernels functions $K(\vec{x}_i, \vec{x}) = \langle \phi(\vec{x}_i), \phi(\vec{x}) \rangle$ leading to a reformulation of the SVM classifier into

$$g(x) = \text{sgn} \left(\sum_{i=1}^n \alpha_i y_i K(\vec{x}_i, \vec{x}) + b \right). \quad (2.36)$$

The standard linear kernel assumed throughout this section is just the dot product so that $K(\vec{x}_i, \vec{x}_j) = \langle \vec{x}_i, \vec{x}_j \rangle$. Other well-known kernels include the polynomial kernel, radial basis functions (RBF), or sigmoid kernels.

While SVMs are formulated for binary classification problems, there are approaches how multi-class classification can be dealt with by SVMs. One approach is known as the *one-vs-rest* mechanism which means that for each of the $|\mathcal{Y}|$ classes a binary classifier is learned distinguishing the class $y' \in \mathcal{Y}$ against all others. According to the *winner-takes-all* paradigm, the class for which $d_{y'}(\vec{x})$ achieves the highest distance wins (Schölkopf and Smola, 2002).

2.2.3 Decision Tree Induction

Another approach to model a ML problem is one where the target function representation is learned by tree induction so that the model is a decision tree. In such a tree, each node corresponds to a test on a specific feature f_j , the branches starting from that node refer to possible values of the feature. Figure 2.4 shows a decision tree for a simple toy problem. Classification is done by descending the tree from top to bottom according to tests at the current node on the specific feature of an observation. Once a leaf is reached, the classification is done. Each path from the root to a leaf constitutes a conjunction of constraints on the features. Overall, a decision tree represents a disjunction of such conjunctions.

Decision trees are one of the most intensively studied approaches to classification learning. Although on complex problems decision trees often produce performance values considerably inferior to the other models discussed in this chapter, they are still often applied because of the intuitive interpretation of the learned model, i.e., the decision tree. In contrast to the other learning approaches described in this chapter, decision trees are capable of learning non-linear functions.

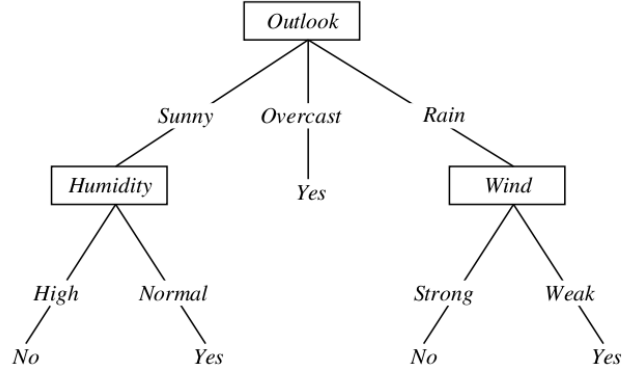


Figure 2.4: Decision tree for the concept *PlayTennis* with the target classes $\mathcal{Y} = \{Yes, No\}$. Figure taken from (Mitchell, 1997, p. 53). A new observation x is classified by sorting it through the tree to the particular leaf node.

Decision trees are typically constructed top-down so that the most effective features are tested in upper nodes. The effectiveness of a feature as possible test for a new branching is assessed by the *information gain*. The information gain $\text{Gain}(S, v_j)$ is the expected reduction of entropy given the feature $f_j(x)$.

$$\text{Gain}(S, v_j) = H(S) - \sum_{t \in \text{values}(f_j(x))} \frac{|S_t|}{|S|} H(S_t) \quad (2.37)$$

where $S \in \mathcal{L}$ is the set of all available training examples which fall in this part of the tree, the set of all possible feature values of f_j is specified by $\text{values}(f_j(x))$, and S_t is the set of all training examples (x, y) where $f_j(x) = t$. $H(S) = \sum_{y' \in \mathcal{Y}} -p_{y'} \log(p_{y'})$ is the entropy over the class distribution where $p_{y'}$ is the proportion of S belonging to class y' . Instead of entropy, other measures for impurity, including the Gini index or classification error, can be applied.

There are various algorithms for the exact process of learning the tree including ID3 and C4.5 as the most prominent ones (Quinlan, 1993). For more details on decision tree learning see (Mitchell, 1997).

2.3 Evaluating Models

This section describes measures and methods to evaluate a fitted model. Evaluation is usually applied to assess the absolute performance of a model or to select the best

predicted class	true class	
	+1	-1
+1	tp	fp
-1	fn	tn

Table 2.1: Contingency table for binary classification scenario.

performing model θ^* out of the set of possible models. Most evaluation measures compare the true target value y with the target value \hat{y} predicted by the classifier. Such a comparison is done on a test set of examples $(x, y) \in \mathcal{T}$. When these examples were used during training so that $\mathcal{T} \subseteq \mathcal{L}$ this is known as *resubstitution*. Resubstitution typically severely underestimates the classifier's generalization error. For a better estimate of the true performance, one should calculate the performance on a separate, held-out test set \mathcal{T} where $\mathcal{L} \cap \mathcal{T} = \emptyset$.

Definition 6 (*Performance on a Held-out Test Set*) The performance reached by a model θ on a held-out test set \mathcal{T} is defined as $\text{perf}(\theta, \mathcal{T})$. Performance estimation is based on a comparison of predicted target values \hat{y} and true target values y .

To accurately estimate the true performance on future data, the data distribution of the test set \mathcal{T} needs to correspond to that of future data. Often, the amount of labeled examples is too limited to guarantee a representative partitioning of the data into training and test set leading to overfitting or erroneous performance estimation. A method known as *k-fold cross-validation* addresses this problem by splitting the data into k partitions. In each fold, another $k-1$ partitions are used for training, and the remaining one for evaluation. Finally, an average over the fold-specific evaluation measures is reported as overall performance. A common setting is $k = 10$.

Comparing the predicted target value \hat{y} and the true target value y of an observation x , four different outcomes are possible: true positives (tp), false positives (fp), false negatives (fn), true negatives (tn). Table 2.1 shows a *contingency* table illustrating the outcomes for a binary classification problem where one class is known as positive and the other as negative. In a non-binary classification problem, a separate contingency table is build for each target class.

Given the numbers from the contingency table, several performance measures can be calculated. The accuracy of a model is defined by

$$ACC = \frac{tp + tn}{tp + tn + fp + fn} \quad (2.38)$$

and accordingly, the error rate is given by

$$ERR = 1 - ACC \quad (2.39)$$

Accuracy and error rate are problematic because these scores do not report on the type of errors made. Other measures have been introduced to circumvent this shortcoming including *Precision*

$$P = \frac{tp}{tp + fp} \quad (2.40)$$

as a measure for the proportion of correct predictions and *Recall*

$$R = \frac{tp}{tp + fn} \quad (2.41)$$

to quantify the number of actual positives identified as such. To combine both measures into a single measure of overall performance, the *F-score* has been introduced (van Rijsbergen, 1979) and is given by

$$F_\alpha = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} \quad (2.42)$$

where α defines the weighting of precision and recall and is often set to $\alpha = 0.5$. This is known as F_1 which simplifies to $F_1 = \frac{2RP}{R+P}$.

When $|\mathcal{Y}| > 2$ two variants of the F-score are usually distinguished. The *micro* F-score is an average over the class-specific F-scores F_y which are each weighted proportionally to the number of examples for this class in the test set

$$F_{\text{micro}} = \sum_{y' \in \mathcal{Y}} \frac{1}{|\mathcal{T}_{y'}|} F_{y'} \quad (2.43)$$

where $\mathcal{T}_{y'} \subset \mathcal{T}$ comprises all examples (x, y) in \mathcal{T} for which $y = y'$. The *macro* F-score, is the unweighted average over the class-specific F-scores:

$$F_{\text{macro}} = \frac{1}{|\mathcal{Y}|} \sum_{y' \in \mathcal{Y}} F_{y'}. \quad (2.44)$$

Chapter 3

Active Learning

In the previous chapter, it was implicitly assumed that at the time of learning, labeled training material would be readily available. In contrast to this *passive* learning scenario, *active* learning is characterized by an interactive and sequential learning scenario where the learner is in control of the data to be made available for training and the learning process is run in iterations of increasingly more data. The objective of active learning (AL) is to select only examples which are useful for training so learning is accelerated and the amount of labeled training material needed to obtain a classifier of a certain performance is reduced.

3.1 Selective Sampling

Early approaches to AL focus on scenarios where unlabeled data is unavailable and are based on *query construction* where examples of high utility are synthesized.¹ Also known as membership queries, this approach is impractical for most real-world applications because human annotators have difficulties labeling synthetically constructed queries. Another shortcoming of query construction is that the synthesized data points are not taken from the underlying data distribution \mathcal{D} and may thus be of limited value in terms of model generalization.

In contrast, *selective sampling* is another, much more widely applied branch of AL where large amounts of unlabeled examples are assumed to be available. Instead of generating examples, examples of high expected utility are selected. Utility here refers to the usefulness of an example for model learning. In the *stream-based* setting, only one example at a time can be accessed and is either rejected or accepted. Stream-based sampling leads to an implicit modeling of the underlying data distribution \mathcal{D} naturally limiting the risk of extensive selection of unrepresentative

¹See (Angluin, 1988) for a theoretically study on query construction and (Ling and Du, 2008) for a recently published approach.

examples such as outliers. In the *pool-based* setting, a large pool of unlabeled examples is readily available allowing to base the selection on the direct comparison of all unlabeled examples. Without an explicit consideration of the data distribution \mathcal{D} during sampling, unrepresentative examples may be selected extensively.

This thesis concentrates on selective sampling as the more realistic scenario for linguistic annotation. Specifically, the pool-based setting is considered which applies well to the scenario of information explosion sketched in Chapter 1 where the number of unlabeled examples, e.g., scientific articles in the biomedical domain, is virtually unlimited. When not explicitly mentioned otherwise, AL refers to pool-based selective sampling henceforth.

Definition 7 (*Pool, Training Set*) A pool $\mathcal{P} \subset \mathcal{X}$ is the set of unlabeled examples $p = (x)$ from which pool-based AL selects. The training set $\mathcal{L} \subset \mathcal{X} \times \mathcal{Y}$ is a set of labeled examples $l = (x, y)$.

It should be noted that AL does not create subsamples of identically and independently distributed (*i.i.d.*) examples. The selected examples are not independently distributed because they are selected with respect to the history of previously selected examples. AL deliberately induces a heavy sampling bias with the distribution of the samples $\mathcal{D}_{\mathcal{L}}$ being very different from the underlying data distribution $\mathcal{D}_{\mathcal{P}}$.

AL, as well as sampling approaches in general, aim at reducing the sample size without loss of model performance. AL can thus be considered an optimization problem with the objective to minimize the size of the subsample $\mathcal{L} \subset \mathcal{P}$ under the constraint of a given target performance.²

Definition 8 (*Optimal Active Learning Strategy*) Given a held-out test set of labeled examples \mathcal{T} , a model $\theta_{\mathcal{L}}$ learned on a set of labeled examples \mathcal{L} , and an aspired target performance perf^* , an AL strategy is optimal if $\text{perf}(\theta, \mathcal{T}) \geq \text{perf}^*$ can be reached with minimal sample complexity $|\mathcal{L}|$.

Instead of searching the *globally* optimal set \mathcal{L} in one step, iterative *greedy* strategies are usually applied selecting one (or multiple) example(s) after another. In each such sampling step a locally optimal selection depending on the history of previous selections is performed with the hope that it will lead to a good global solution. This form of myopia is a standard assumption in active learning.

²In practise, the ultimate objective is the reduction of *real* annotation cost needed, instead of the number of examples, to yield the target performance. This issue is addressed in Chapter 11.

The core of any AL approach is a function $u(p)$ used to assess the utility of an example $p \in P$ for model induction. AL selection is based on this utility score. According to Definition 8, the optimality of an AL approach is described with respect to a test set \mathcal{T} representing the target population and its distribution $\mathcal{D}_{\mathcal{T}}$. The utility function thus ideally consists of the following two components:

- The *informativeness* $I(p, \theta)$ which quantifies the contribution of an example p in terms of model improvement when added to the training set \mathcal{L} , and
- the *representativeness* $R(p, \theta)$ which quantifies how representative an example p given the data distribution $\mathcal{D}_{\mathcal{T}}$ is.

While informativeness rewards an example's ability to improve the model θ trained on all training data \mathcal{L} available at that point, representativeness attempts to avoid the selection of outliers irrelevant in the target distribution $\mathcal{D}_{\mathcal{T}}$ to which the final model should be applied.

Definition 9 (Utility function) A utility function $u(p, \theta) = \phi(I(p, \theta), R(p, \theta))$ estimates the utility of an example $p \in \mathcal{P}$ by a function on the informativeness and the representativeness of p .³

In each iteration, greedy AL then selects the example

$$p^* = \operatorname{argmax}_{p' \in \mathcal{P}} u(p', \theta). \quad (3.1)$$

Such a selected example for which the true label y is required is called a *query*. The label is queried from an omniscient *oracle* or *teacher* which in practise is a human annotator. After labeling, $l^* = (x, y)$ is added to the training set and the next AL starts. Algorithm 1 formally describes this general framework for greedy AL.

3.2 Approaches to Greedy Active Learning

Approaches to greedy AL can be distinguished by the definition of the utility function, the mode of data access (pool-based, stream-based, or even query generation), the use of a single or multiple classifiers for selection, the dependency of the approach

³We here consider only combinations of informativeness and representativeness on a low-level combination manner and not as two separate criteria. This issue is discussed in detail in Chapter 8 which explicitly deals with AL for multiple criteria.

Algorithm 1 General Framework for Greedy Active Learning

Given:

\mathcal{L} : set of labeled examples $l = (x, y) \in \mathcal{X} \times \mathcal{Y}$

\mathcal{P} : set of unlabeled examples $p = (x) \in \mathcal{X}$

$T(\mathcal{L})$: a learning algorithm

$u(p, \theta)$: utility function

Algorithm:

loop until stopping criterion is met

1. learn model: $\theta \leftarrow T(\mathcal{L})$
2. select $p^* \leftarrow \operatorname{argmax}_{p' \in \mathcal{P}} u(p', \theta)$
3. query label y for p^* : $l^* \leftarrow (x, y)$
4. $\mathcal{L} \leftarrow \mathcal{L} \cup \{l^*\}$, $\mathcal{P} \leftarrow \mathcal{P} \setminus \{p^*\}$

return $\mathcal{L}^* \leftarrow \mathcal{L}$ and $\theta^* \leftarrow T(\mathcal{L}^*)$

to a specific learning algorithm, and the learning task applied to (classification, regression, and confidence estimation or ranking). This section describes and categorizes well-known approaches to greedy AL by their definition of the utility function.⁴ Remarkably, most approaches define the utility function just as a function of the informativeness completely ignoring representativeness.

3.2.1 Statistically Optimal Active Learning

Statistically optimal approaches to AL employ a utility function which is based on the objective function to which the model is fitted and by which the learning is eventually evaluated. This is typically the estimated generalization error according to the particular loss function. As a result, the utility of an example x is then proportional to the reduction of loss expected when this example is added to \mathcal{L} . Statistically optimal approaches to AL explicitly attempt to quantify the difference of the performance between the ideal and the current classifier.

3.2.1.1 Reduction of Expected Generalization Error

Explicit reduction of the expected generalization error as an approach to AL was first described by Roy and McCallum (2001). In an AL iteration, for each unlabeled example $p = (x)$, all tuples (x, y') , with $y' \in \mathcal{Y}$, are consecutively added to the training set and the label-specific expected error $\tilde{E}_{\mathcal{L} \cup \{(x, y')\}}$ given this new training

⁴With slightly different foci, Olsson (2009) and Settles (2009b) have recently also published reviews of approaches to AL.

set is estimated. Depending on the particular loss function, the error is calculated in different ways. Roy and McCallum (2001) formulate the errors for the 0/1- and the log-loss. The overall expected error $\tilde{E}_{\mathcal{L} \cup \{x\}}$ for an unlabeled example $p = (x)$ is then the average over the label-specific errors weighted by their posteriors in \mathcal{L} . The corresponding utility function is defined as

$$u_{\text{GE}}(p, \theta) = \frac{1}{|\mathcal{Y}|} \sum_{y' \in \mathcal{Y}} \tilde{E}_{\mathcal{L} \cup \{(x, y')\}} \cdot P_{\mathcal{L}}(y'|x). \quad (3.2)$$

For HMMs, Anderson and Moore (2005) proposed several objective functions for the HMM problems “state learning”, “path learning”, “model learning”, and “classification”. Similarly to (Roy and McCallum, 2001), the utility function is defined as the *value of information*, i.e., the expected reduction of loss when an example $p = (x)$ would be added to \mathcal{L} .

3.2.1.2 Minimization of Expected Variance

Earlier approaches towards statistically optimal AL did not directly optimize the generalization error but the classifier’s variance, instead. The mean squared error (MSE) of a model can be decomposed into bias and variance so that $\text{MSE} = \text{Bias}^2 + \text{Variance}$. According to the *Bias-Variance Dilemma* (Geman et al., 1992), a model with a low bias has a large variance and vice versa.⁵ Assuming an approximately unbiased learner, minimizing the error equals minimizing the learner’s variance. According to this intuition, Cohn et al. (1996) proposed an approach to AL based on the attempt to minimize the learner’s variance. Similarly, as described above for the expected error reduction, the estimated variance \tilde{V} for an example $p = (x)$ is the weighted average over the label-specific variance estimates. Accordingly, the utility function is defined as

$$u_{\text{VAR}}(p, \theta) = \frac{1}{|\mathcal{Y}|} \sum_{y' \in \mathcal{Y}} \tilde{V}_{\mathcal{L} \cup \{(x, y')\}} \cdot P_{\theta}(y'|x). \quad (3.3)$$

Modeling of the expected variance over the input space requires knowledge of the example distribution and the possibility to calculate the classifier’s variance in a closed form. Cohn et al. (1996) applied this approach to AL for neural networks, Gaussian mixture models, and locally-weighted linear regression where closed-form calculation can be done as they show.⁶ For arbitrary learning algorithms, such a closed-form is, however, not possible rendering this approach impractical for many applications.

⁵When other loss functions hold, another decomposition may hold (Hansen and Heskes, 2000).

⁶Note that Cohn et al. (1996) apply their approach to a query construction scenario.

Instead of modeling the global variance, Saar-Tsechansky and Provost (2004) estimate the local variance for each example $p \in \mathcal{P}$ separately, allowing the use of arbitrary learning algorithms. The local variance is found empirically as the mean observed variance on a set of classifiers $\mathcal{C} = \{\theta_1, \dots, \theta_e\}$ learned from subsamples of \mathcal{L} . Experiments were performed for the task of class-probability estimation.

Another approach related to variance reduction is based on the maximization of the Fisher information (Zhang and Oles, 2000). The Fisher information measures the amount of information, an observation x carries about an unknown parameter λ_j , i.e., the impact x has on the efficiency of parameter estimation. For a MaxEnt model, the Fisher information $\mathcal{I}(\lambda_j)$ is defined as the sum over all possible labels $y' \in \mathcal{Y}$ over the partial derivations of the maximum likelihood estimate by λ_j :

$$\mathcal{I}_x(\lambda_j) = \sum_{y' \in \mathcal{Y}} P(y'|x) \frac{\partial^2}{\partial \lambda_j^2} \log \left(P(y'|x) \right). \quad (3.4)$$

For $\theta = (\lambda_1, \dots, \lambda_k)$, the Fisher information has the form of a $k \times k$ matrix $\mathcal{I}(\theta)$, where the values of the diagonal correspond to the Fisher information values defined in Equation 3.4 for the parameters λ_j . The asymptotic information value of a particular unlabeled example $p = (x) \in \mathcal{P}$ is calculated by the Fisher information ratio

$$\mathcal{F}_{x,\mathcal{P}}(\theta) = \text{tr}(\mathcal{I}_x(\theta)^{-1} \mathcal{I}_{\mathcal{P}}(\theta)) = \sum_{j=1}^k \frac{\sum_{x' \in \mathcal{P}} P(x') \mathcal{I}_{x'}(\lambda_j)}{P(x) \mathcal{I}_x(\lambda_j)} \quad (3.5)$$

where $\mathcal{I}_x(\theta)$ is the Fisher information matrix for example $p = (x)$, $\mathcal{I}_{\mathcal{P}}(\theta)$ is the respective matrix for all examples in the pool \mathcal{P} , and the trace tr is the sum of the elements along the principal diagonal of the resulting matrix. The Fisher information value can be interpreted as the asymptotic reduction of model uncertainty by querying example p . Accordingly, the utility function is defined as

$$u_{\text{FIR}(p,\theta)} = -\mathcal{F}_{x,\mathcal{P}}(\theta) \quad (3.6)$$

so that examples with lower Fisher information ratio are preferentially selected. The Fisher information ratio has been applied as a utility function for AL by Hoi et al. (2006) and Settles and Craven (2008).

Statistically optimal approaches share high computational costs as a common disadvantage. In practise, only when the labeling costs of single examples are extremely high, application of these approaches is justifiable. Naïve implementations of error reduction approaches are often extremely inefficient because for all examples $p \in \mathcal{P}$ a model needs to be trained to estimate the new loss. Roy and McCallum (2001)

proposed several optimizations and approximations to speed up selection: (a) testing only a subsampling the examples $P' \subset \mathcal{P}$, (b) subsampling from \mathcal{L} to calculate the expected error only on a subsample of \mathcal{P} , (c) incremental model training, and (d) re-classification. However, the statistical optimality might be lost when these tricks and simplifications are applied.

3.2.2 Expected Model Change

Examples that when incorporated into the training data would lead to a major model change, may be considered as highly influential. Another strand of AL approaches focuses on the expected model change as a measure for utility. Maximizing expected model change is an approach to minimize future generalization error.

In context of maximum margin classification, and especially SVMs, an elegant approach to AL has been developed by Schohn and Cohn (2000) and Tong and Koller (2000). The utility of an example $p = (x)$ is measured by its proximity to the hyperplane $d(\vec{x})$ given the model parameters \vec{w} and b (cf. Equation 2.33):

$$u_{\text{SVM}}(p, \vec{w}, b) = -|d(\vec{x})| \quad (3.7)$$

An example that lies within the margin, i.e., for which $|d(\vec{x})| < 1$ holds, effects the selection of support vectors and by this the decision function learned. The calculation of the distance of an example to the hyperplane is computationally inexpensive because it is only a dot product computation.

Gradient-based strategies are often applied to determine model parameters by optimizing the respective objective functions. This is, for example, the case for the MaxEnt model and CRFs as discussed in Chapter 2 where the objective function is based on the log-likelihood $\ell(\mathcal{L}, \theta)$ of a probability distribution. When adding a new example (x, y) to the training set \mathcal{L} , the change in the gradient can be taken as an approximation to the change in the model. Let $\nabla \ell(\mathcal{L}, \theta)$ be the current gradient and $\nabla \ell(\mathcal{L} \cup \{(x, y)\}, \theta)$ the gradient obtained when adding the new example. Settles et al. (2007) propose the *expected gradient length* (EGL) as utility function

$$u_{\text{EGL}}(p, \theta) = \sum_{y' \in \mathcal{Y}} P_{\theta}(y'|x) \nabla \ell(\mathcal{L} \cup \{(x, y')\}, \theta) \quad (3.8)$$

which sums over all possible labels because the true label for an example is unknown at query time. According to this utility function, the example which imparts the greatest model change would be selected.

3.2.3 Uncertainty Sampling

Uncertainty sampling (US) is a relatively simple approach to AL where the utility of an example is based on the uncertainty, as the inverse of the confidence, of the current classifier in its prediction (Lewis and Gale, 1994). US is model-independent and can be combined with any passive learner that returns confidence or probability estimates for its predictions.

Several measures to quantify uncertainty have been proposed. The *least-confidence* (LC) utility function is based on the a-posteriori of the most likely label y' for an example $p = (x)$ and is defined as

$$u_{\text{LC}}(p, \theta) = 1 - \max_{y' \in \mathcal{Y}} P_{\theta}(y'|x). \quad (3.9)$$

In the case of binary classification, this metric comprises all available information on the distribution of the a-posteriori probabilities. However, in a multi-class scenario, information about the posteriors of the other labels is lost. Scheffer and Wrobel (2001) argued that the confidence distribution over the labels should be incorporated to get a better uncertainty estimate. A simple approach to this is to consider the margin between the best and the second best label. A small margin means that the decision between the best and second best label is hard. The *margin* (MA) utility function is defined as

$$u_{\text{MA}}(p, \theta) = -\left(\max_{y' \in \mathcal{Y}} P_{\theta}(y'|x) - \max_{\substack{y'' \in \mathcal{Y} \\ y' \neq y''}} P_{\theta}(y''|x)\right). \quad (3.10)$$

A third variant considers the posterior distribution of all possible labels. The *entropy* (ENT) utility function is defined as

$$u_{\text{ENT}}(p, \theta) = \sum_{y' \in \mathcal{Y}} P_{\theta}(y'|x) \cdot \log(P_{\theta}(y'|x)). \quad (3.11)$$

Variants of these utility functions exist for specific tasks and learning scenarios, such as the tree entropy for syntactic parsing (Hwa, 2000), and variants for sequence classification (see Section 4.2.3).

US is prone to selecting outliers or unrepresentative examples as these tend to exhibit a high uncertainty. But when added to the training material, these examples do not improve the classifier's performance (Roy and McCallum, 2001). An important advantage of US is its low computational complexity compared to statistically optimal approaches. Despite these shortcomings, empirical studies found good performance for US in practical use (Laws and Schütze, 2008; Settles and Craven, 2008).

3.2.4 Query-by-Committee

Seung et al. (1992) describe the *Query-by-Committee* (QbC) framework where the utility score of an example is derived from the disagreement within a committee of classifiers $\mathcal{C} = \{\theta_1, \dots, \theta_e\}$. The original QbC framework has a committee of $|\mathcal{C}| = 2$ and is defined for a stream-based setting. An example on which the predictions of the two committee members diverge is to be selected. QbC – at least in theory – exhibits faster convergence of the base learner’s performance compared to US. This is so because QbC selects controversial examples and is thus less likely to spend much effort exploring outliers (Freund et al., 1997).

Several variations of this original QbC approach have been proposed with the main difference in the mechanism to sample the committee of classifiers and measures for divergence. Moreover, variants of QbC are often applied to pool-based settings where the degree of divergence is interpreted as an utility score. In the following, common approaches to committee sampling and divergence metrics are discussed.

3.2.4.1 Sampling the Committee

The original QbC framework has a strong foundation in computational learning theory and is based on the intuition of version space reduction. The version space V is the subset of all hypotheses h_i from the hypothesis space H which are consistent with the training set \mathcal{L} , i.e., hypotheses that correctly classify all examples in \mathcal{L} so that $h(x) = y$, where y is the correct label for x (Mitchell, 1997). Each unseen example added to \mathcal{L} potentially reduces the number of hypotheses consistent with \mathcal{L} . QbC aims at choosing the example that reduces the version space the most. Under ideal conditions, QbC even halves the version space (see Section 3.4). Seung et al. (1992) constructed the committee by randomly sampling two hypotheses from the version space.

Depending on the actual learning problem, data, and the dimension of the feature space, the version space may be huge. In many real-world machine learning problems, the feature space is extremely high-dimensional, so that sampling the version space is practically impossible. Several hundreds of thousands of features are common in language and speech processing. To make QbC applicable to real-world problems, Gilad-Bachrach et al. (2006) projected the version space to a lower dimensional space by intersecting the version space on \mathcal{L} with the version space on the unlabeled example $p = (x)$. With respect to p it is sufficient to sample from this smaller version space V' .

In general, the notion of consistent hypotheses does not apply to probabilistic classifiers where model parameters need to be estimated so that they statistically best fit the training data \mathcal{L} . Instead of sampling hypotheses, Dagan and Engelson (1995) proposed to construct the committee by randomly sampling committee members θ_i . Under the assumption that the model parameters $\theta = (\lambda_1, \dots, \lambda_k)$ are mutually independent, committee members θ_i can be constructed by sampling each model parameter λ_j from the posterior distribution $P(\lambda_j|\mathcal{L})$ separately. This requires the distribution $P(\lambda_j|\mathcal{L})$ which Dagan and Engelson approximated by a truncated normal distribution. Similarly, McCallum and Nigam (1998) sampled the model parameters for a NB learner from a Dirichlet distribution instead of the normal distribution.

Several heuristics and practical implementations to sample committees in a computationally less expensive manner have been proposed. Abe and Mamitsuka (1998) proposed *Query-by-Bagging* (QbB), where each committee member is trained on a random subsample of \mathcal{L} , and *Query-by-Boosting* which is based on AdaBoost to create the committee members more intelligently. AdaBoost (Freund and Schapire, 1996) iteratively adds examples which previously were misclassified to some initial training set to learn a new committee member. Interestingly, the more complex Query-by-Boosting does not consistently outperform Query-by-Bagging. AL with multiple views, known as *Co-Testing*, was proposed by Muslea et al. (2000). All committee members represent redundant views on \mathcal{L} , i.e., disjoint feature vectors, which could independently be used for classification. Finding proper feature vector splits in practical applications is one of the major problems of this approach.

QbC also has its roots in ensemble learning, which is about combining the outputs of multiple classifiers to obtain improved performance. In ensemble learning, a certain level of diversity among the committee members is a key property for good committees (Krogh and Vedelsby, 1995). According to this intuition, Melville and Mooney (2003) proposed DECORATE, an approach to produce highly diverse committees by adding artificially constructed examples to the training set. As an extension, ACTIVE-DECORATE successfully employs such diverse committees in an AL scenario (Melville and Mooney, 2004).

Besides reduced complexity in committee sampling, another advantage of the above mentioned heuristics is that they can be thought of as wrapper mechanisms around an underlying passive learning algorithm which is treated as a black box. In this way, heuristics to QbC can easily be applied with arbitrary learning algorithms.

3.2.4.2 Divergence Measures

Engelson and Dagan (1996) proposed the *Vote Entropy* (VE) as a measure of dis-

agreement based on the entropy of the distribution of the predicted labels of the committee members $\theta_i \in \mathcal{C}$.

$$u_{\text{VE}}(p, \mathcal{C}) = - \sum_{y' \in \mathcal{Y}} \frac{V(y', x)}{|\mathcal{C}|} \log \frac{V(y', x)}{|\mathcal{C}|} \quad (3.12)$$

where $V(y', x)$ denotes the number of committee members θ_i according to which $g_{\theta_i}(x) = y'$ on example $p = (x)$. To directly focus on examples which tend to improve upon the F-score, Ngai and Yarowsky (2000) proposed the *F-complement* (FC) where the relative, pairwise F-scores between all members $\theta_i \in \mathcal{C}$ of the committee are compared so that

$$u_{\text{FC}}(p, \mathcal{C}) = \frac{1}{2} \sum_{\theta_i \in \mathcal{C}} \sum_{\substack{\theta_j \in \mathcal{C} \\ i \neq j}} \left(1 - F(g_{\theta_i}(x), g_{\theta_j}(x)) \right) \quad (3.13)$$

where $F(g_{\theta_i}(x), g_{\theta_j}(x))$ is the F-score of the prediction of g_{θ_j} relative to the prediction of g_{θ_i} for example $p = (x)$.

When class membership probabilities are also available as a result of classification, they may be incorporated into the divergence score. In the most straight-forward manner, the utility function already discussed in the context of US (Equations 3.9, 3.10, and 3.11) can also be applied here (Körner and Wrobel, 2006). A prerequisite is the availability of a combined distribution over the probability estimates for the committee. A simple approach to this is the mean over the conditional distributions of all committee members $P_{\theta_i}(y|x)$ so that committee's distribution is

$$P_{\text{avg}}(y|x) = \frac{1}{|\mathcal{C}|} \sum_{\theta_i \in \mathcal{C}} P_{\theta_i}(y|x). \quad (3.14)$$

Now, the least-confidence, the margin, and the entropy metric from the previous section can be applied by inserting the committee's distribution $P_{\text{avg}}(y|x)$. This application of the originally US-based utility functions in the QbC scenario does not well fit the idea of divergence but is instead based on group consensus.

Another way to incorporate knowledge about the committee member's class distributions is based on the Kullback-Leibler (KL) divergence D (Kullback and Leibler, 1951) which quantifies the difference of two probability distributions P_{θ_1} and P_{θ_2} :

$$D(P_{\theta_1}(y) || P_{\theta_2}(y)) = \sum_{y' \in \mathcal{Y}} P_{\theta_1}(y') \log \frac{P_{\theta_1}(y')}{P_{\theta_2}(y')}.$$

The *KL divergence to the mean* (KLM) is the average KL divergence between each distribution $P_{\theta_i \in \mathcal{C}}(y|x)$ and $P_{\text{avg}}(y|x)$. The KLM divergence is a special form of the

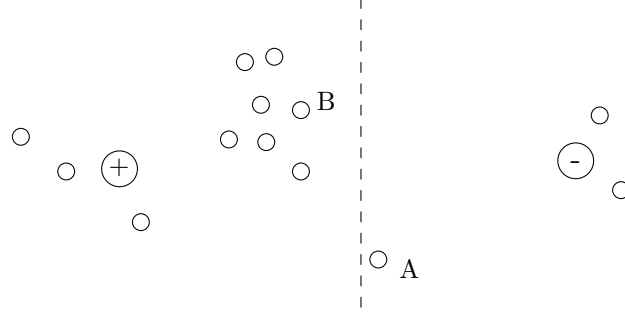


Figure 3.1: Most informative example (A) is globally less useful compared to the highly representative, but less informative example (B). Dashed line represents current decision boundary. Small circles are unlabeled examples, big circles labeled ones.

Jensen–Shannon divergence (Lin, 1991) where the single distributions are associated with importance weights. A high KLM score indicates that the distributions diverge considerably. A utility function based on the KLM divergence is defined as

$$u_{\text{KLM}}(p, \mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{\theta_i \in \mathcal{C}} D(P_{\theta_i}(y|x) || P_{\text{avg}}(y|x)). \quad (3.15)$$

KLM has been successfully applied in context of AL by McCallum and Nigam (1998) and Melville and Mooney (2004).

3.2.5 Representativeness-aware Approaches

As AL is about learning a good classifier with minimal label complexity, considering informative examples is reasonable. However, selection based on informativeness only can lead to the selection of examples which are unrepresentative, i.e., located in regions of low density, or are even outliers. Both can result in degraded classifier performance when evaluated on a test set \mathcal{T} . Figure 3.1 intuitively illustrates this problem (adopted from Guo and Greiner (2007)). According to an informativeness criterion, example A would be selected (it is extremely close to the current decision boundary). Example B would be a much better choice as it is surrounded by a number of other examples so that the model would learn more knowing B.

As evidenced by the previous sections, most approaches to AL do not consider representativeness in the utility function. Note, however, that density-weighted or distribution-sensitive sampling is only necessary in pool-based sampling. Stream-based sampling, in contrast, naturally keeps the underlying data distribution.

Guo and Greiner (2007) proposed an approach to representativeness-aware AL. The underlying idea is to select the example $p = (x)$ which – when added to the training data \mathcal{L} – provides maximum mutual information about labels of the remaining unlabeled examples of \mathcal{P} . Since the true label for p is unknown at sampling time, the target y' which helps maximizing the mutual information most is assumed the best label for $p = (x)$. The utility function is thus given by

$$u(p, \mathcal{L}) = - \min_{y' \in \mathcal{Y}} \sum_{x'' \in \mathcal{P}} H_{\theta_{\mathcal{L} \cup \{(x, y')\}}}(\hat{y} | x'') \quad (3.16)$$

where $\theta_{\mathcal{L} \cup \{(x, y')\}} = T(\mathcal{L} \cup \{(x, y')\})$ is the model learned when adding x with the “best” label y' to \mathcal{L} . Furthermore, $H_{\theta}(\hat{y} | x) = - \sum_{y' \in \mathcal{Y}} P_{\theta}(y' | x) \log P_{\theta}(y' | x)$ represents the conditional entropy of the unknown label \hat{y} with respect to x and the model θ .

This approach is highly related to the approaches discussed in Section 3.2.1 – with the difference that the density in \mathcal{P} is considered here to incorporate representativeness. Here, representativeness and informativeness are interwoven on a low level and thus define a monolithic utility function. Often, however, representativeness and informativeness are considered as two different criteria and so do not fall under the definition of a utility function as given in this chapter. Instead, those approaches should rather be considered a multi-criteria AL scenario which is defined and discussed in detail in Chapter 8.

3.3 Adaptations for Feasibility

In the general formulation of greedy AL (Algorithm 1), one example is selected in each AL iteration. Each such selection step, requires expensive parameter estimation because the utility functions are based on the prediction or confidence scores provided by a trained classifier. As a result, approaches to AL where only a single example per iteration is selected are hardly feasible in practise due to long selection times resulting from model retraining and utility assessment per example. The selection of a so-called *batch* set \mathcal{B} of examples with $|\mathcal{B}| > 1$ per AL iteration has been proposed and applied in many studies as a remedy. *Batch-mode* AL selects a certain number of examples with the highest utility scores instead of one example, only.

However, batch-mode AL gives rise to a new issue, namely the diversity among the selected examples. When not explicitly controlled, the batch might consist of highly similar examples which get similar utility scores. In this case, the overall utility of the complete batch is presumably much lower than the sum of the single examples’ utility scores so that repetitive selection and, by this, annotation of identical or

highly similar examples is not beneficial for learning. Several approaches to control the diversity of the batch have been proposed (Brinker, 2003; Shen et al., 2004) and are discussed in the context of multi-criteria AL in Chapter 8.

Heuristic approaches to AL, namely QbC and US, are usually computationally less complex than the statistically optimal approaches which often require repetitive estimation of model parameters θ for each example to be tested. However, also for QbC and US a model needs to be trained in each AL iteration. To further speed up this process, Lewis and Catlett (1994) proposed *heterogeneous* US where the learning algorithm used during AL differs from the target learning algorithm which is used to learn the final model from all data \mathcal{L}^* made available through AL. In their specific experiment, they applied a probabilistic classifier during selection time which trained much faster than the decision tree learner applied to obtain the final model. While they reported positive findings, this approach has hardly been readopted by others. Chapter 7 studies applicability and limitations of heterogeneous AL scenarios.

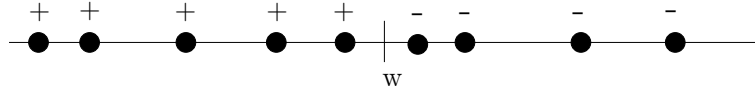
3.4 Sampling Complexity Bounds

Several works in computational learning theory have analyzed theoretical labeling complexity bounds for AL in general and specific approaches to it (Cohn et al., 1994; Freund et al., 1997; Dasgupta, 2004, 2006; Kääriäinen, 2006). The estimation of sampling complexity bounds is typically formulated dependent on an error rate ε . Given an example x randomly drawn from the underlying data distribution \mathcal{D} and a hypothesis h , the error of this hypothesis is defined as the probability that $\hat{y} = h(x)$ disagrees with the true label y :

$$\text{error}_{\mathcal{D}}(h(x), y) = p(h(x) \neq y) \quad (3.17)$$

Computational learning theory teaches that $\text{error}_{\mathcal{D}}(h(x), y) \leq \varepsilon$ can be achieved with $m = \mathcal{O}(\frac{1}{\varepsilon})$ randomly drawn examples (Mitchell, 1997). This holds only if the data is separable by the hypothesis h .

In specific settings, an AL strategy can exponentially reduce the sampling complexity to an upper bound of $m = \mathcal{O}(\log(\frac{1}{\varepsilon}))$. This has been nicely illustrated by Dasgupta (2004) on a binary toy example of finding a linear separator in \mathbb{R} where the data lies on a real line and there is a point $w \in \mathbb{R}$ which separates the positive from the negative class. See Figure 3.2 for a visualization. Binary search reduces the sampling complexity to $m = \mathcal{O}(\log(\frac{1}{\varepsilon}))$ compared to $m = \mathcal{O}(\frac{1}{\varepsilon})$ obtained by random sampling. An exponential reduction of the sampling complexity for binary classification has also been shown for the original QbC framework by Freund et al. (1997) where every new example added to \mathcal{L} is assumed to halve the version space (cf. Section 3.2.4).

Figure 3.2: Toy example for finding a linear separator in \mathbb{R} .

Such great savings, however, are only achievable under the *realizability assumption* which means that for a given target function correct classification of all training and test examples is possible. Moreover, the absence of labeling noise is assumed. Generalization of the reported complexity bound is highly problematic so that the actual benefit of AL depends upon the particular hypothesis class and the pool of unlabeled examples. Finally, the lower bound of sample complexity for AL is given by $\Omega(\frac{1}{\epsilon})$ while the upper bound is given by $\mathcal{O}(\log(\frac{1}{\epsilon}))$ (Dasgupta, 2004, 2006). Note, however, that in general AL can also perform significantly worse than random sampling. This holds especially in the case of non-realizability. As for noise, bounded rate class noise with a noise rate $\alpha < \frac{1}{2}$ still allows exponential savings of sampling complexity when simple noise-cancelling methods based on repeated queries are applied (Kääriäinen, 2006). However, this does not hold for systematic noise persistent over repeated queries which presumably is a much more realistic scenario. Although exponential savings are unrealistic in practise, AL can still exhibit lower sampling complexity than passive learning.

3.5 Related Fields of Research

AL in general as well as particular approaches to it are related to several other fields of research, the most important ones of which this section briefly reviews. From a learning perspective, AL is related to semi-supervised learning (Chapelle et al., 2006). Both approaches aim at reducing the number of *manually* labeled examples: AL by selectively sampling and semi-supervised learning by making use of unlabeled data which are assumed to be available at no costs.

AL, at least the selective sampling strand of it, can be also considered a subsampling strategy.⁷ Sampling strategies are usually motivated by the reduction of computational complexity in context of inductive learning due to poor scalability characteristics of many learning algorithms as well as slow access times to data stored in data bases. AL, in contrast, focuses on the reduction of sampling complexity, mainly to alleviate the burden of human annotation. Despite contrary motivation, AL shares many characteristics and methods such as approaches to sequential sampling. As

⁷The interested reader should refer to Scholz (2007) for a good overview of sampling strategies.

such, AL is highly related to *progressive sampling* where increasingly larger samples are used for training as long as model performance increases (Provost et al., 1999).

From a statistician’s point of view, AL may be described as a form of experimental design which is about the acquisition of data through experiments (Atkinson and Donev, 1992). In particular, *optimal* experimental design aims at generating small samples still sufficient to estimate parameters with the same precision as would be achieved by a naïve, non-optimal experimental design with much larger samples. The main motivation for optimal experimental design is to reduce the financial costs of experiments. Some approaches to AL are formulated in this spirit (Schein, 2005).

A decision theorist might consider AL as the problem of finding an optimal decision given several alternatives and a utility function to assess them. Finding an optimal decision is then an optimization problem. Due to feasibility, AL is usually solved by greedy optimization algorithms finding a locally optimal decision at each step.

3.6 Applications of Active Learning

AL has found application to a range of problems addressed by learning-based systems. Especially for scenarios where large amounts of unlabeled data is available at no or relatively low costs but creation of labeled training data is a costly procedure, AL is a promising solution to cost reduction. As such, it found application to several problems of robotics including path planning (Zhang and Kim, 1997), vision capabilities (Salganicoff et al., 1996), and object grasp control (Morales et al., 2004). Another lively field of application are multimedia retrieval systems (Tong and Chang, 2001; Huang et al., 2008; Wang et al., 2009). Moreover, AL was found useful in bioinformatics where Liu (2004) applied it to gene expression data for cancer classification and Danziger et al. (2007) made use of AL to optimize experimental data choices for more rapid discovery of biological function through experiments.

AL has also attracted lots of attention in the field of NLP. This is evidenced by the workshop on “Active Learning for Natural Language Processing” held in conjunction with the 2009 Conference of the North American Chapter of the Association for Computational Linguistics. AL has been successfully applied to a range of tasks including text classification (Lewis and Gale, 1994; McCallum and Nigam, 1998), part-of-speech tagging and chunking (Engelson and Dagan, 1996; Ngai and Yarowsky, 2000; Ringger et al., 2007), statistical parsing (Hwa, 2000), Named Entity Recognition (Shen et al., 2004; Tomanek et al., 2007a), and lately even statistical machine translation (Haffari et al., 2009). Riccardi and Hakkani-Tür (2005) demonstrate how AL helps speed up training classifiers for speech and audio processing. For a more detailed review, refer to (Olsson, 2009) and Chapter 4.

3.7 Summary

This chapter has introduced and discussed AL as a selective sampling strategy with the primary motivation of AL to reduce the sampling complexity to achieve a particular classifier performance. A special focus has been set to pool-based AL because this settings corresponds well to the prevalent situation faced nowadays in most annotation campaigns: large amounts of *unlabeled* texts are readily available at extremely low costs. The annotation, however, is a costly process.

In this chapter, sampling complexity is measured by the number of labeled examples. As we will see in Chapters 4 and 11, this notion of sampling complexity has to be adapted when AL is applied to real-world annotation problems. Moreover, while this chapter provided a general and task-independent introduction to AL, the next chapter elaborates in detail on the specific approach to AL used throughout the rest of this thesis for economizing on labeling costs for linguistic annotation.

Part II

Towards Resource-Aware Annotation through Active Learning

Chapter 4

Active Learning for Named Entity Recognition

This chapter briefly introduces the NLP task of Named Entity Recognition (NER) and describes our approach to it. Subsequently, we specify our AL framework in the context of this task and discuss task-specific design decisions in detail. Moreover, several utility functions for AL, which will be used throughout this thesis, are adapted to the NER scenario. The second half of this chapter serves to evaluate our NER approach both in a *passive* and an *active* learning setting. Passive learning results serve as the upper bound of performance that is attainable on the given data by a particular learner. Performance of AL is studied in greater detail, including a comparison of the utility functions and an analysis of the characteristics of the resulting samples.

The NER task should be understood as a sample application scenario from the greater domain of NLP. This task is employed throughout the rest of the thesis as a test bed to evaluate the different approaches and modifications to AL. We chose the NER task for two reasons: firstly, the research in for this thesis was initially motivated by a research project¹ where large amounts of named entity annotations were needed. Secondly, NER is a task that is subject to major entity type changes whenever it comes to its application in new domains, fields, or genres. As a result, recurring annotation endeavours are a common scenario, so that the availability of strategies to make annotation less resource-intensive implies high practical gains. Thirdly, NER is a crucial component in IE systems where recognized entity mentions are typically the input to relation or fact extraction methods. IE systems experience increasing application to real-world problems so that practical NER applied to new problems (and so the creation of new training material) will be a real need in the near future. This shows the relevance of the experiments of this thesis.

¹The STEMNET project aimed at designing and constructing a knowledge management system for the field of stem cell biology. More information on the particular annotation endeavours undertaken as part of this project is given in Section [12.2](#).

In [Colorado], [Mark Thompson] threw an eight-hitter for his third complete game and [Ellis Burks] homered and drove in three runs as the [Colorado Rockies] beat the [Pittsburgh Pirates] 9-3.

Figure 4.1: A sample sentence taken from the CoNLL corpus with NE annotations. Colors: [location], [organization], [person].

4.1 Named Entity Recognition

The Message Understanding Conferences, as well as other competitions in the field of NLP, were originally founded with the goal of promoting and evaluating research in language processing. NER was defined as a separate analysis step in NLP during the preparation of the sixth Message Understanding Conference (MUC-6) in 1995. In contrast, earlier Message Understanding Conferences had considered IE as a single monolithic task subsuming NER (Grishman and Sundheim, 1996).

Today, NER is an integral part of most IE systems. It comprises the identification and classification of textual expressions that refer to *Named Entities* (NEs). These entities are named instances of a specific type, such as textual mentions of city or country names which may be classified by the type *location*. It should be noted that the term “Named Entity” has no precise linguistic definition, but was by-and-large coined from an application point of view, *viz.* NLP (Nadeau and Sekine, 2007). This lack of a precise definition may be one reason why reaching agreement on what constitutes an NE in a particular scenario is usually a very time-consuming and challenging undertaking.

For the analysis of newspaper language, common NE types include persons, locations, and organizations. Figure 4.1 shows a text snippet where NEs of these types are highlighted. When applied to other genres or domains, other entities are of interest, such as genes and proteins, cell types, or organisms in scientific articles of the biomedical domain.

4.1.1 Previous Work

NER has been intensively studied during the last decade. This was partly initiated by a series of shared tasks and competitions, such as MUC (Marsh and Perzanowski, 1998), CoNLL (Tjong Kim Sang and De Meulder, 2003), BioCreative (Hirschman et al., 2005), and JNLPBA (Kim et al., 2004) which were performed for different languages, domains, and text genres. Due to the overwhelming amount of literature in this field, we only report on a few early and trend-setting works. Early works on

NER tended to focus on rule-based systems (Black et al., 1998; Fukuda et al., 1998), whereas current, state-of-the-art systems are mostly based on machine learning (or hybrid) approaches. For a comprehensive overview of NER from 1991 to the present day see Nadeau and Sekine (2007).

Most of the models described in Chapter 2 have been applied to NER. Bikel et al. (1997) employed a slightly modified HMM, Borthwick et al. (1998) presented one of the first applications of the MaxEnt model to NER, Isozaki and Kazawa (2002) described a NER system based on SVMs. More recently, CRFs have found their way into and are now a de-facto standard for NER (McCallum and Li, 2003).

There is still a lot of ongoing research on approaches to NER, mostly focusing on the application of NER systems developed and tested on the general newspaper domain in English to other languages (Wu et al., 2003; Vijayakrishna and Sobha, 2008; Benajiba et al., 2008) or domains (Klinger et al., 2008; Iria, 2009; Zhao and Liu, 2008) and studying ways to reduce the burden of creating large amounts of training material. These undertakings demonstrate that NER is a relevant task, making it an important and realistic application scenario for our studies on AL.

4.1.2 Sequence Labeling Problem

The words in a natural language text are not an arbitrary accumulation – but their order and composition is important and grammatical constraints hold. In this thesis, we use the term “token” to refer to a word which is recognized as such in a sentence. Some NLP tasks, including part-of-speech (POS) tagging, chunking, and NER, can be considered as segmentation tasks, which means that text must be divided into meaningful segments that constitute subsequences of words from the original text. Moreover, these segments are often categorized by, for example, different entity or POS types. Consequently, we treat NER as a sequence labeling problem and employ a (first-order linear-chain) CRF as our approach to NER.

We employ a rich set of standard token-level features for NER.² These include the word itself, various orthographic features such as capitalization, the occurrence of special characters such as hyphens, suffixes and prefixes, and context information in terms of features of neighboring tokens to the left and right of the current token. Table 4.1 provides an overview of the features used. These features are very suitable and general enough to be used in most (sub)domains for entity recognition. Moreover, there has been discussion that CRFs are able to handle such large amounts of presumably highly correlated features.

²See Nadeau and Sekine (2007) for a detailed description of features typically used for NER.

feature class	description
orthographic	based on regular expressions (<code>HasDash</code> , <code>IsUpperCase</code> , ...) and a transformation rule: capital letters replaced by “A”, lowercase letters by “a”, digits by “0” (example: <code>IL2</code> \rightarrow <code>AA0</code> , <code>have</code> \rightarrow <code>aaaa</code>)
lexical	prefix and suffix of length 3, stemmed form of each token
contextual	features of neighboring tokens (one to the left, one to the right) to model local context

Table 4.1: Standard feature set used for CRF-based NER.

Many works on NER, especially in the biomedical domain, have shown that the performance of a CRF model can be immensely increased when this standard feature set is optimized and extended in an appropriate way (Klinger et al., 2008). However, throughout this thesis we employ the same standard feature set for comparability of all experiments. That is because the focus of this thesis is not on feature selection and outperforming state-of-the-art performance of NER when given huge amounts of training data, but on cost-efficient ways to provide highly useful training material. Our standard feature set does, though, yield respectable performance values as shown below in Section 4.4.1.

Another reason why we refrain from a fine-tuned feature set in our experiments is that feature selection assumes the availability of labeled training material which, in practise, is rarely the case. Its creation would be inconsistent with the goal of AL to decrease annotation effort. Thus, a more authentic scenario is to run AL with a core feature set such as ours and then fine-tune it once data is available. In this spirit, we intentionally avoid using features such as semantic triggers words (Zhou et al., 2005), references to external dictionaries (Klinger et al., 2007), or POS tags because they are highly dependent on the specific subdomain and entity types used.

4.2 Active Learning

Based upon the formulation of a general framework for greedy AL in the previous chapter, we specify here an AL framework for the specific scenario of NER.

4.2.1 Related Work

This section lists other approaches to AL for segmentation tasks. Remarkably, none of the papers listed below have applied any of the statistically optimal AL approaches. Instead, so-called *heuristic* approaches, *viz.* approximations to QbC and US, or expected model change were applied. As discussed in the previous chapter, statistically optimal approaches are often not appropriate in practise due to their computational complexity and dependency on specific learning algorithms.

As for QbC, the most common divergence metrics used are the Vote Entropy, the KL divergence, and the F-complement. Committees were mostly built by the simple bagging mechanism. QbC has been applied to POS tagging (Engelson and Dagan, 1996; Ringger et al., 2007), Chunking (Ngai and Yarowsky, 2000), and NER (Tomanek et al., 2007a; Olsson, 2008; Settles and Craven, 2008). As for US-based AL, different forms of confidence scores and distributions have been used in utility functions, including token- and sequence-level confidence scores as well as complete and k-best confidence distributions.³ US has been applied to POS tagging (Ringger et al., 2007) and NER (Kristjansson et al., 2004; Laws and Schütze, 2008; Settles and Craven, 2008). In the above-mentioned works the models applied were mostly variants of MaxEnt models and CRFs. As for approaches based on model change, Shen et al. (2004) applies AL for Support Vector Machines based on the proximity to the hyperplane, and Settles and Craven (2008) apply the Expected Gradient Length utility function. In both cases, experiments were performed for the NER task.

This short overview reveals that, despite the differences, these works share many principle design decisions. An approach that is different in spirit is the BootMark method presented by Olsson (2008). BootMark selects complete documents instead of much smaller granularities because its main focus is on the creation of completely annotated *documents* instead of non-consecutive training material.

For a more comprehensive description of approaches to AL for NLP, please refer to the literature review by Olsson (2009).

4.2.2 NER-specific AL Framework

This section describes the framework we use for application of AL to the NER task. This framework is a customization of the general framework of greedy AL, as described in Algorithm 1 on page 32. In the following description, the task-specific design decisions of the customized framework are discussed.

³K-best confidence distributions are applied when the number of possible labels or label sequences is large so that the complete distribution of confidence scores would be too expensive to compute.

If not mentioned otherwise, the CRF with the features described previously in this chapter is applied as default model, and our approaches to AL are all implemented as a wrapper treating the model as a black box subroutine. Batch-mode selection considerably reduces computational complexity of AL-based selection and, as a result, human idle time when waiting for the next examples for annotation. In accordance with Requirement 2, batch-model AL is applied throughout the rest of this thesis if not mentioned otherwise.

In the general framework of greedy AL, an example $p = (x)$ is an atomic “unit” – self-sufficient and independent from other examples – subject to classification. In the NER scenario, as well as for other segmentation tasks, such a unit is not naturally available. We have already argued that NER is best addressed as a sequence labeling problem due to the interdependence of single tokens. So, what is then an appropriate sequence size for AL selection?

From a linguistic and a human annotation perspective, one may argue that an appropriate sequence should be a self-contained phrase (in accordance with the task at hand) such as a noun phrase, a sentence, or even a paragraph.⁴ An overly short phrase may not cover the complete entity (or other kinds of relevant segments) and thus complicate human annotation decisions. Moreover, due to the overly limited context available in short phrases, annotation of selected phrases requires intensive access to context words. This presumably largely increases annotation time. From an ML perspective, we would like a sequence of reasonable length, i.e., not too small as it would lack sequence characteristics. In contrast, from a selective sampling perspective, shorter sequences may lead to more precise utility scores and do thus allow for more focused selection.

Several selection granularities have been proposed in the literature on AL for segmentation tasks. Laws and Schütze (2008) select single tokens, Olsson (2008), in strong contrast, sets the selection granularity to the document level, and most approaches to AL for segmentation tasks have focused on a sentence-level selection granularity (Engelson and Dagan, 1996; Ringger et al., 2007; Tomanek et al., 2007a). While the token-level selection granularity exhibits the steepest and most rapidly converging learning curves, from the annotation perspective it is highly questionable whether the annotation of isolated tokens is feasible for human annotators. On the other hand, from a selective sampling perspective, it is questionable whether the document-level is an appropriate selection granularity, as the human annotator may waste time on labeling useless subsequences. As shown by Olsson (2008), it is difficult to improve upon random selection with document-level selection.

⁴A noun phrase is a phrase whose head is either a noun or a pronoun (Jurafsky and Martin, 2000).

Algorithm 2 NER-specific Active Learning Framework**Given:** b : number of examples to be selected in each iteration \mathcal{L} : set of labeled examples $l = (\vec{x}, \vec{y}) \in \mathcal{X}^n \times \mathcal{Y}^n$ \mathcal{P} : set of unlabeled examples $p = (\vec{x}) \in \mathcal{X}^n$ $T(\mathcal{L})$: a learning algorithm $u(p, \theta)$: utility function**Algorithm:**

loop until stopping criterion is met

1. learn model: $\theta \leftarrow T(\mathcal{L})$
2. sort $p \in \mathcal{P}$: let $S \leftarrow (p_1, \dots, p_m) : u(p_i, \theta) \geq u(p_{i+1}, \theta), i \in [1, m], p \in \mathcal{P}$
3. select b examples p_i with highest utility from S : $\mathcal{B} \leftarrow \{p_1, \dots, p_b\}, b \leq m, p_i \in S$
4. query labels for all $p \in \mathcal{B}$: $\mathcal{B}' \leftarrow \{l_1, \dots, l_b\}$
5. $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{B}', \mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{B}$

return $\mathcal{L}^* \leftarrow \mathcal{L}$ and $\theta^* \leftarrow T(\mathcal{L}^*)$

In this thesis, we focus on a sentence-level selection granularity which is a reasonable unit from a linguistic perspective, contains enough context for many segmentation tasks, and allows for acceptably focused selection. In the context of AL for NER, we define an example to be a sequence $\vec{x} = (x_1, \dots, x_n)$ of n tokens. An unlabeled example is given by $p = (\vec{x})$ and a labeled one by $l = (\vec{x}, \vec{y})$.

Algorithm 2 formalizes the NER-specific AL framework for an arbitrary utility function $u(p, \theta)$. This AL framework differs from the general framework of greedy AL (Algorithm 1) only by the fact that now in each iteration multiple examples are selected and that these examples are sequences.

Although specified for a scenario where a single model is employed in the utility function, adaptation to a committee of models is straightforward and affects only Steps 1 and 2 in the loop of Algorithm 2. As specified here, the algorithm returns both the selected and annotated sample \mathcal{L}^* as well as the model θ^* . Often, as a result of AL, we are more interested in \mathcal{L}^* from which we intend to learn a final model. We thus distinguish the models employed during AL for selection purposes and those induced later from the created data \mathcal{L}^* .

Definition 10 (*Selector*) A model θ employed during AL is called selector. QbC-based AL employs a committee \mathcal{C} with several selectors $\mathcal{C} = \{\theta_1, \dots, \theta_e\}$.

Definition 11 (*Consumer*) The model θ^* induced from the sample \mathcal{L}^* obtained by the AL process is called consumer.

4.2.3 Utility Functions

In this thesis, QbC and US-based approaches to AL are applied. We deliberately decided against the use of statistically optimal approaches to AL due to their high computational complexity, which is not compatible with Requirement 2 postulating low selection time. This holds especially when AL is to be applied to NLP tasks which are characterized by extremely high-dimensional feature spaces and therefore complex learning problems. As an example, for NER on the 15,875 sentences of the CoNLL corpus (see below), the feature space has over 100,000 dimensions.

An advantage of QbC over US is its strong foundation in computational learning theory and the fact that under certain conditions (cf. Section 3.4) sampling complexity can be exponentially reduced. This property, however, was shown for the original version of QbC which is based on hypothesis sampling (Seung et al., 1992). For heuristics to QbC, including Query-by-Bagging used in this thesis, there are no such estimates. US is appealing due to its lower computational complexity compared to QbC, as with US, only one model instead of a committee of models needs to be trained in each iteration of AL.

The NER-specific AL framework with a sentence-level selection granularity requires adaptations of the utility functions, which in the previous chapter were formulated for examples consisting of a single element to be classified. We group the utility functions used throughout this thesis in three categories. *Divergence-based* functions include utility functions formulated for variants of QbC and based on explicit measurement of the divergence amongst the committee members. *Sequence confidence-based* and *token confidence-based* utility functions refer to those formulated in the context of the US approach, where the confidence scores or the distribution over confidence scores is calculated either on the sentence or the token level, respectively.

The utility functions formulated in Chapter 3 do not handle sequence information directly but instead are calculated on the token level and then aggregated over the sentence. Several methods of aggregation have been discussed in literature, including the mean average, the minimum/maximum, and the standard deviation over token-level utility scores (Olsson, 2008). Experiments have shown that aggregation by the mean average performs best (Lichtenwald, 2009). This aggregation is given by

$$u^{\bar{s}}(p, \theta) = \frac{1}{n} \sum_{i=1}^n u(x_i)$$

where $p = (\vec{x})$ and x_i is the token at position i in a sentence \vec{x} of length n .

Note, that utility functions $u_{\text{name}}^{\bar{s}}$ refer to aggregates over token-level scores, while utility functions u_{name}^s refer to scores calculated on the complete sequence directly.

Divergence-based Utility Functions The Vote Entropy (Equation 3.12) and the F-complement (Equation 3.13) utility functions are here based on the predicted Viterbi sequence $\vec{y}^* = g_\theta(\vec{x})$ so that the adapted versions for our NER-specific AL framework are defined as

$$u_{\text{VE}}^{\bar{s}}(p, \mathcal{C}) = -\frac{1}{n} \sum_{i=1}^n \sum_{y' \in \mathcal{Y}} \frac{V(y', i)}{|\mathcal{C}|} \log \left(\frac{V(y', i)}{|\mathcal{C}|} \right) \quad (4.1)$$

$$u_{\text{FC}}^s(p, \mathcal{C}) = \frac{1}{2} \sum_{\theta_i \in \mathcal{C}} \sum_{\substack{\theta_j \in \mathcal{C} \\ i \neq j}} \left(1 - F(g_{\theta_j}(\vec{x}), g_{\theta_i}(\vec{x})) \right) \quad (4.2)$$

where $V(y', i)$ is the number of committee members which predicted the label y' for position i in \vec{x} . In u_{FC}^s , $F(g_{\theta_j}(\vec{x}), g_{\theta_i}(\vec{x}))$ indicates the F-score between the predicted sequences of both committee members θ_i and θ_j . The KL divergence to the mean (Equation 3.15) is also based on the entropy over the token-level confidence distribution $P_{\theta_j}(y_i|x)$ of each committee member θ_j and the respective utility function are defined as

$$u_{\text{KLM}}^{\bar{s}}(p, \mathcal{C}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{|\mathcal{C}|} \sum_{\theta_j \in \mathcal{C}} D(P_{\theta_j}(y_i|x) || P_{\text{avg}}(y_i|x)). \quad (4.3)$$

Token Confidence-based Utility Functions We here calculate the utility score on each position i in the observation sequence \vec{x} by the marginal probability (Equation 2.31) of the respective label y' from the Viterbi sequence \vec{y}^* . These token-level utility scores are then aggregated to obtain the respective adaptations of the least confidence (LC, Equation 3.9), the margin (MA, Equation 3.10), and the entropy (ENT, Equation 3.11) utility function:

$$u_{\text{LC}}^{\bar{s}}(p, \theta) = 1 - \frac{1}{n} \sum_{i=1}^n \max_{y' \in \mathcal{Y}} P_\theta(y_i = y'|x) \quad (4.4)$$

$$u_{\text{MA}}^{\bar{s}}(p, \theta) = -\frac{1}{n} \sum_{i=1}^n \left(\max_{y' \in \mathcal{Y}} P_\theta(y_i = y'|x) - \max_{\substack{y'' \in \mathcal{Y} \\ y' \neq y''}} P_\theta(y_i = y''|x) \right) \quad (4.5)$$

$$u_{\text{ENT}}^{\bar{s}}(p, \theta) = \frac{1}{n} \sum_{i=1}^n \sum_{y' \in \mathcal{Y}} P_\theta(y_i = y'|x) \cdot \log(P_\theta(y_i = y'|x)) \quad (4.6)$$

Sequence Confidence-based Utility Functions With sequence-level confidence estimates based on the conditional probability of the Viterbi sequence given the observation sequence (Equation 2.24), the sequence confidence-based utility functions

based on LC and MA are given by

$$u_{\text{LC}}^s(p, \theta) = 1 - P_{\theta}(\vec{y}^* | \vec{x}) \quad (4.7)$$

$$u_{\text{MA}}^s(p, \theta) = -(P_{\theta}(\vec{y}^* | \vec{x}) - P_{\theta}(\vec{y}^{**} | \vec{x})) \quad (4.8)$$

where \vec{y}^* and \vec{y}^{**} are the first- and second-best Viterbi sequences, respectively. The entropy utility function is omitted due to the huge number of possible label sequences. In the next section, the sampling complexity and efficiency as well as sample characteristics of these eight utility functions are analysed.

4.3 General Experimental Settings

We here evaluate the utility functions described above in the NER-specific AL framework. We performed such a comparative evaluation of utility functions because of a lack of comprehensive studies providing generalizable results on which utility function performs best. Although two other comparative studies of approaches to AL have been published recently (Olsson, 2008; Settles and Craven, 2008), it is questionable whether their findings generalize to our particular scenario.⁵

In all experiments, the learner is based on CRFs with the same feature set as described above. This holds both for the selector and the consumer. We apply an implementation of CRFs as found in the machine learning toolkit Mallet (McCallum, 2002). For committee-based AL, Query-by-Bagging (QbB) is applied throughout this thesis. For QbB, the single committee members are trained on a random sample of $\frac{|\mathcal{C}|-1}{|\mathcal{C}|}$ sentences drawn without replacement from \mathcal{L} . If not mentioned otherwise, $|\mathcal{C}| = 3$ in order to keep the computational cost at a minimum. The experimental settings described above are reused throughout the rest of the thesis if not mentioned otherwise.

4.3.1 Evaluation Measures

AL approaches are usually evaluated by the sampling complexity, i.e., the number of examples needed to yield a specific model performance. Typical performance

⁵Olsson (2008) comprehensively compares different approaches to QbC and US in the context of the BootMark method where complete documents are selected. It is unclear in how much the specific selection granularity affected the results. With a focus on sequence labeling and NER, Settles and Craven (2008) compare several approaches to AL; however, generalizing their results to our setting is problematic due to another cost measure being applied: While we consider the number of tokens being annotated, Settles and Craven consider the number of sentences instead. As sentences are subject to a high variability in length (cf. Figure 11.1), costs based on tokens cannot be directly compared with costs based on sentences.

measures are the accuracy or the F-score yielded. We consider the F-score as our ultimate performance measure. If not mentioned otherwise, the term “F-score” henceforth refers to the micro F_1 -score as described in Section 2.3. When applied to segmentation tasks, Recall and Precision are calculated by

$$R = \frac{\# \text{ of correct segments found}}{\# \text{ of all segments in text}} \quad (4.9)$$

$$P = \frac{\# \text{ of correct segments found}}{\# \text{ of segments found}} \quad (4.10)$$

so that the segment F-score is calculated as in Equation 2.42 (assuming $\alpha = 0.5$) but based on the above formulations of recall and precision (Jurafsky and Martin, 2000). In the NER scenario, a segment corresponds to an entity mention. Tokens labeled as the non-entity class do not count during this evaluation.

Annotation Cost Measure The label complexity, also called annotation effort, is quantified by some kind of *cost measure*. Traditionally, studies on AL assume a unit cost per selected example. However, when selecting complete sentences, the unit cost assumption is overly simplistic as sentences tend to vary enormously in their length (cf. Table 4.2). Moreover, it has been shown that the actual choice of the cost measure strongly influences the quantification of the success of a particular AL approach (Claire et al., 2005; Haertel et al., 2008a; Settles et al., 2008). Our cost measure is based on the number of annotated tokens making up the sentences contained in \mathcal{L} . We assume uniform costs $c = 1$ for all tokens so that

$$\text{cost}(\mathcal{L}) = \sum_{(\vec{x}, \vec{y}) \in \mathcal{L}} \sum_{i=1}^n c \quad (4.11)$$

where n is the length of the sequence. We believe this to be a reasonable approximation of annotation effort in the absence of an empirically more adequate task-specific model for annotation cost.

Learning Curves The relation of label complexity and model performance can be visualized by a learning curve showing the performance as a function of annotation effort (see Figure 4.2). For the learning curve, the performance of the model $\theta_{\mathcal{L}_j}$ induced from \mathcal{L}_j is calculated for increasingly larger subsets $\mathcal{L}_j \subseteq \mathcal{L}$. Sampling-based evaluation techniques such as cross-validation cannot be applied here because the examples $l \in \mathcal{L}$ are subject to a heavy sampling bias.

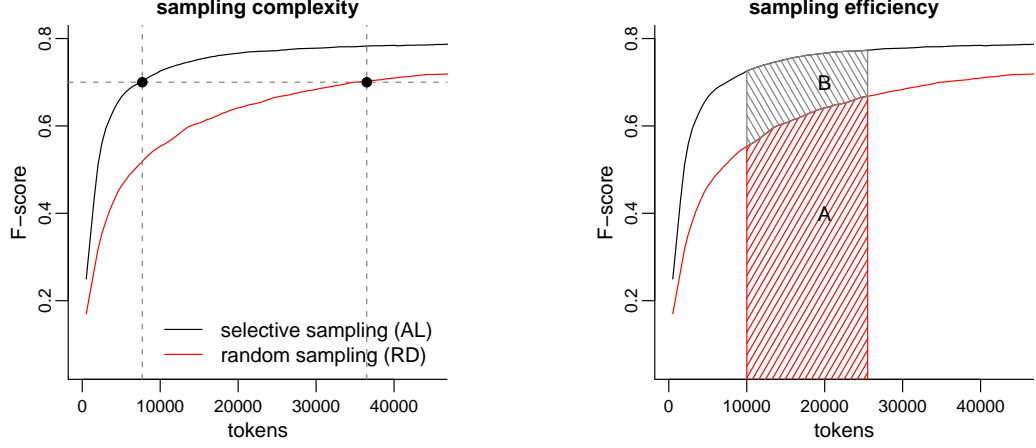


Figure 4.2: Sampling complexity for a target performance of $F=0.7$: $CFP_{0.7}(AL) = 7,700$ tokens and $CFP_{0.7}(RD) = 36,000$ tokens. Sampling efficiency in interval $[10,000; 25,000]$ is calculated by RAI measure as $\frac{B}{A} - 1$.

Sampling Complexity and Sampling Efficiency To evaluate the performance of sampling strategies, we consider *sampling complexity* as well as *sampling efficiency*. Sampling complexity describes the number of examples needed to yield a particular target performance $perf(\theta_{\mathcal{L}}, \mathcal{T})$ on the held-out test set \mathcal{T} where $\theta_{\mathcal{L}}$ specifies the model induced from the actively obtained sample \mathcal{L} . We propose the *Cost for Target Performance* (CFP) measure as an operationalization of sampling complexity. CFP quantifies the cost, according to an arbitrary cost measure, needed to obtain a target performance F^* given a sampling strategy S :

$$CFP_{F^*}(S) = \underset{cost(\mathcal{L}_j)}{\operatorname{argmin}} perf(\theta_{\mathcal{L}_j}, \mathcal{T}) \geq F^*. \quad (4.12)$$

Figure 4.2 visualizes the CFP measure for two sampling strategies.

Sampling efficiency describes how well a model performs on a particular sample. We calculate sampling efficiency on several sample size positions to level out outliers. The *Area Under the Learning Curve* (AUC) in the interval $[a, b]$ is given by

$$AUC(S, a, b) = \int_{j=a}^b perf(\theta_{\mathcal{L}_j}, \mathcal{T}) dj \quad (4.13)$$

where the integral is approximated by adding the performance at step-wise increased values of j . The efficiency of a particular sampling strategy depends to a great extent on the performance of the baseline approach. We define the *Relative Area Increase* (RAI) of the AUC of the selective sampling approach over the baseline's AUC by:

$$\text{RAI}(S_{\text{AL}}, S_{\text{base}}, a, b) = \left(\frac{\text{AUC}(S_{\text{AL}}, a, b)}{\text{AUC}(S_{\text{base}}, a, b)} - 1 \right) \cdot 100. \quad (4.14)$$

The RAI measure assesses the relative increase of sampling efficiency over a baseline sampling strategy. Our RAI measure is similar to the cost reduction measure presented by Haertel et al. (2008a) with the difference that Haertel et al. considered cost reduction on single positions while we chose an interval to level out local variations. Figure 4.2 visualizes the RAI measure.

Throughout the rest of this thesis, CFP and RAI are used as operationalizations of sampling complexity and sampling efficiency. It should be noted, that CFP and RAI might suggest inconsistent conclusions as to which utility function performs better. Moreover, RAI and CFP depend greatly on the evaluation granularity applied, i.e., the number of tokens by which \mathcal{L}_j increases. In this chapter, \mathcal{L}_j is increased so that additional 500 tokens are contained in each evaluation step.

4.3.2 Corpora

We tested all our approaches on four common entity-annotated corpora to see whether the same tendencies can be observed in different scenarios. Two corpora are from the newswire and two other corpora are from the biomedical domain. Both from the newswire and from the biomedical domain there is a corpus with three entity classes and one with considerably more classes.

From the general-language newspaper domain, we took the MUC7 corpus (Linguistic Data Consortium, 2001) as well as the English data set of the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003). MUC7 consists of three independent parts and for our experiments we took the training part only. These documents are New York Times articles from 1996 about airplane crashes. MUC7 has annotations of seven different entity types: *persons*, *organizations*, *locations*, *times*, *dates*, *monetary expressions*, and *percentages*. The CoNLL corpus consists of a collection of Reuters newswire articles on international politics, sports, and finance, and is annotated with three entity types: *persons*, *locations*, *organizations*, and *misc*. From the original CoNLL corpus, we removed the *misc* annotations as they appeared inconsistent.

From the sublanguage biology domain we used the oncology part of the PENNBIOIE corpus (Kulick et al., 2004), which consists of some 1150 PubMed abstracts. Originally, this corpus contains gene, variation event, and malignancy entity annotations. For our simulations, we built two sub-corpora by filtering out entity annotations. The gene sub-corpus (PBGENE) contains annotation of the three gene entity types

	CONLL	MUC7	PBGENE	PBVAR
number of tokens	241,952	78,329	267,320	267,320
class distribution over tokens	0.36	0.18	0.16	0.07
number of sentences	15,785	3,022	10,570	10,570
avg. sentence length	15.3	25.9	25.3	25.3
sentence length interval	[3,49]	[3,71]	[3,64]	[3,64]
number of entity types	3	7	3	6
number of entity mentions	23,365	5,144	12,958	6,261
avg. entity length	1.48	1.61	1.34	1.48
avg. no. of entity mentions per sent.	1.48	1.70	1.23	0.59
entity tokens	34,622	8,283	17,379	9,247
avg. no. of entity tokens per sent.	2.2	2.74	1.6	0.9

Table 4.2: Characteristics of the simulation corpora

generic, *protein*, and *rna*, while the variation event sub-corpus (PBVAR) is built from the annotations of the variation entity types *type*, *event*, *location*, *state-altered*, *state-generic*, and *state-original*. Thus, PBGENE and PBVAR are based on the same sentences but contain different entity annotations.

Table 4.2 provides descriptive corpus statistics. It shows the size of the corpora in terms of tokens and sentences.⁶ The class distribution is the entropy over the token class a-priori probabilities. The classes include the entity classes as well as a class called OUTSIDE in NER which indicates that a token is not part of an entity mention. The entropy is normalized to [0,1]. PBVAR has the most biased distribution and CONLL the least. The table also informs about the overall number of entity mentions, the number of entity mentions per sentence, and the average entity length measured by the number of tokens in the entity mentions. While the average entity length is quite similar in all corpora, the number of entity mentions and entity tokens per sentence deviates considerably in the PBVAR corpus: sentences are more sparsely populated with entities than in the other corpora.

⁶We removed sentences of considerable over and under length (beyond +/- 3 standard deviations around the average sentence length as well as all sentences shorter than 3 tokens) from all four corpora, so that the numbers reported here may differ from those cited in the original sources.

4.3.3 Randomizations

For all our experiments, we repeat the runs to obtain a statistically meaningful result. AL runs are always started from a random seed set. If not mentioned otherwise, we always took the same seed sets in all our experiments. These seed sets consist of 20 randomly sampled sentences. All results reported are averages over the single runs. If not mentioned otherwise, 20 independent runs are performed.

For the single runs we split our corpora into a *pool* $\mathcal{P} \subset \mathcal{X}^n$ to select from, and a *test* $\mathcal{T} \subset \mathcal{X}^n \times \mathcal{Y}^n$ to generate learning curves. It is important to mention, that all splits were made just once. So all experiments are based on the same collection of splits and seed sets for comparability. The original CoNLL corpus has an explicit evaluation and training set. These sets were used in the competition, so we also use these sets as a pool and test set for comparability of results. Thus, for CoNLL we have just one split, in all our individual runs so that only the seed sets are varied. On MUC7, PBVAR, and PBGENE, we split the complete corpus in pools (90% of the sentences) and complementary test sets (remaining 10%).

4.4 Results

4.4.1 Passive Learning

The passive learning performance of our CRF-based approach to NER is compared to that of other relevant NER systems on the same corpora to assess whether it achieves reasonable results. Furthermore, the passive learning performance theoretically constitutes an upper bound for the performance achievable with AL. While previous work, especially on AL for SVMs, has reported that classifier performance yielded on samples obtained by AL is sometimes higher than the performance of a classifier trained on the whole data set (Schohn and Cohn, 2000; Ertekin et al., 2007), we could not observe this behavior in our experiments.

To obtain performance values comparable to the ones yielded by our experiments on AL, we performed passive learning runs on the same splits used in AL and averaged over the single runs. While on MUC7, PBGENE, and PBVAR this means that 20 independent passive learning runs were performed and averaged, on CoNLL the passive learning performance is based on one run only.

Table 4.3 lists the average F-scores. The performance values of the single runs deviate slightly, the highest standard deviation can be observed on the PBVAR corpus

	CoNLL	MUC7	PBGENE	PBVAR
F-score	0.83	0.88 ± 0.014	0.83 ± 0.008	0.78 ± 0.023

Table 4.3: Performance (F-score and standard deviation) of passive learning obtain by 10-fold cross-validation. On the CoNLL corpus, no cross-validation was performed because there is a designated training and evaluation set.

– presumably because of the high variance of entity mentions contained per sentence. With an F-score of about 83%, our approach to NER ranges in the midfield of the systems that took part in the CoNLL-2003 competition (English); the best two system there achieved F-scores of around 88% (Tjong Kim Sang and De Meulder, 2003). On the MUC7 corpus, we achieved an F-score of about 88%; the best performing systems of the MUC7 competition yielded F-scores in the low 90s (Marsh and Perzanowski, 1998). Direct comparison of the results is especially problematic here because the evaluation scenario was very dissimilar.⁷ On PBGENE, we achieve a passive learning F-score of 83%, a CRF with a specialized feature set, feature induction, and lots of domain knowledge was reported to yield about 86% F (McDonald and Pereira, 2005). Similarly, on PBVAR a specialized NER system was reported to achieve an F-score of about 82% (McDonald et al., 2004), while our unspecialized, task-independent approach to NER yields an F-score of about 78%.

One should be careful in directly comparing the results reported for other systems with ours for two reasons: firstly, we do not use exactly the same corpora or evaluation scenario (different forms of cross-validation or train/test splits, removal of sentences with over-/underlength in our scenario), and secondly, we applied an untuned approach to NER, i.e., in its default configuration with a standard feature set, whereas for the systems reported on a lot of engineering and fine-tuning was carried out to obtain the performance scores published. However, this juxtaposition of results shows that our CRF-based approach to NER achieves respectable results on all corpora in its default configuration.

4.4.2 Active Learning

This section empirically evaluates different utility functions in our NER-specific AL scenario (cf. Section 4.2.3) in terms of sampling complexity and sampling efficiency. Based on this comparative study, three utility functions – one from each of the

⁷Groups participating in MUC7 evaluated their systems against an official test set, while in our experiments we employed the MUC7 training set in a cross-validation manner.

utility function category	utility function	MUC7 F = 0.85	CONLL F = 0.78	PBGENE F = 0.78	PBVAR F = 0.77
divergence-based	$u_{VE}^{\bar{s}}$	18,519	25,014	34,514	23,518
	u_{FC}^s	26,014	38,010	42,513	45,519
	$u_{KLM}^{\bar{s}}$	23,016	32,506	45,016	28,019
sequence confidence-based	u_{LC}^s	18,516	44,011	32,011	26,014
	u_{MA}^s	19,020	33,013	33,013	28,512
token confidence-based	$u_{LC}^{\bar{s}}$	17,516	24,011	30,013	22,517
	$u_{MA}^{\bar{s}}$	19,014	24,509	31,018	22,511
	$u_{ENT}^{\bar{s}}$	17,518	25,013	30,015	27,011
random sampling	RD	37,518	47,009	83,011	123,390

Table 4.4: Sampling complexity for different utility functions given by CFP scores according to a corpus-specific target performance (F). The best-performing utility function per category is highlighted (see Section 4.2.3 for the definition of the utility functions).

utility function categories “divergence-based”, “token confidence-based”, and “sequence confidence-based” – are selected for further use in this thesis. Additionally, characteristics of the resulting samples are analyzed to understand better why some utility function perform better than others and which are appropriate scenarios for the application of selective sampling. Finally, we briefly discuss the appropriateness of the chosen selection granularity.

4.4.2.1 Evaluation of Utility Functions

Instead of plotting learning curves for all utility functions, we report on sampling complexity in terms of CFP scores and on sampling efficiency in terms of RAI scores. For the CFP scores, a corpus-specific target performance F^* is chosen so as to be as large as possible, with the constraints that (a) it should occur before the convergence phase and (b) so that all metrics reach this score within a maximum of 50,000 tokens.⁸ Random sampling (RD) is taken as a baseline scenario for the RAI score which we calculate on the interval of [10000, 30000] tokens. This interval efficiently excludes the very first start-up phase as well as the convergence phase.

Tables 4.4 and 4.5 show the respective CFP and RAI scores. With the exception of the RAI score of u_{LC}^s on CONLL, all metrics clearly outperform random sampling.

⁸Due to variations in the learning curves, we define that the specified target performance must have been reached in at least 3 successive evaluation positions to avoid erroneous performance assessment due to singular positive outliers.

utility function category	utility function	MUC7	CoNLL	PBGENE	PBVAR
divergence-based	$u_{VE}^{\bar{s}}$	6.07	5.51	8.15	20.40
	u_{FC}^s	2.63	0.79	6.32	16.60
	$u_{KLM}^{\bar{s}}$	4.24	3.26	4.71	18.90
sequence confidence-based	u_{LC}^s	6.26	-2.91	8.65	19.40
	u_{MA}^s	6.00	1.13	8.31	19.50
token confidence-based	$u_{LC}^{\bar{s}}$	6.50	6.01	9.62	20.80
	$u_{MA}^{\bar{s}}$	6.32	6.05	8.92	20.70
	$u_{ENT}^{\bar{s}}$	6.39	5.98	9.77	20.00

Table 4.5: Sampling efficiency of different utility functions given by RAI scores over random selection in the interval of [10000, 30000] tokens. The best-performing utility function per category is highlighted (see Section 4.2.3 for the definition of the utility functions).

Amongst the divergence-based utility functions, $u_{VE}^{\bar{s}}$ performs best on all corpora, both in terms of CFP and RAI scores, and overall u_{FC}^s is the worst-performing utility function. Interestingly, $u_{KLM}^{\bar{s}}$ performs considerably worse than all other utility functions (except u_{FC}^s). This is especially surprising when compared to $u_{VE}^{\bar{s}}$, for example, which calculates utility scores based only on the predicted labels while $u_{KLM}^{\bar{s}}$ takes the distributions of confidence scores into account and may thus be assumed to perform much better.

As for US-based utility functions, performance differences are less clear. Amongst the sequence confidence-based utility functions, u_{LC}^s and u_{MA}^s perform similarly and amongst the token confidence-based utility scores there is also no clear winner. These performance characteristics hold for both the CFP and the RAI scores on all of our four corpora, so that they may be generalized for a wider range of NER-scenarios.

We chose the following three utility functions for further application in this thesis: $u_{VE}^{\bar{s}}$ as a representative for the divergence-based utility functions, u_{LC}^s for the sequence confidence-based utility functions, and $u_{MA}^{\bar{s}}$ for the token confidence-based utility functions. The choice of $u_{VE}^{\bar{s}}$ was motivated simply by its considerably better sampling complexity and sampling efficiency compared to u_{FC}^s and $u_{KLM}^{\bar{s}}$. We chose u_{LC}^s because of its lower computational complexity compared to u_{MA}^s (only the best Viterbi sequence needs to be determined). Finally, we preferred $u_{MA}^{\bar{s}}$ over $u_{ENT}^{\bar{s}}$ because of slightly lower computational complexity and over $u_{LC}^{\bar{s}}$ because $u_{MA}^{\bar{s}}$ has a better theoretical motivation (cf. Section 3.2.3 and (Scheffer and Wrobel, 2001)).

Figure 4.3 shows the learning curves for AL with these three utility functions. In direct comparison, $u_{MA}^{\bar{s}}$ performs best but is closely followed by $u_{VE}^{\bar{s}}$ and u_{LC}^s , which

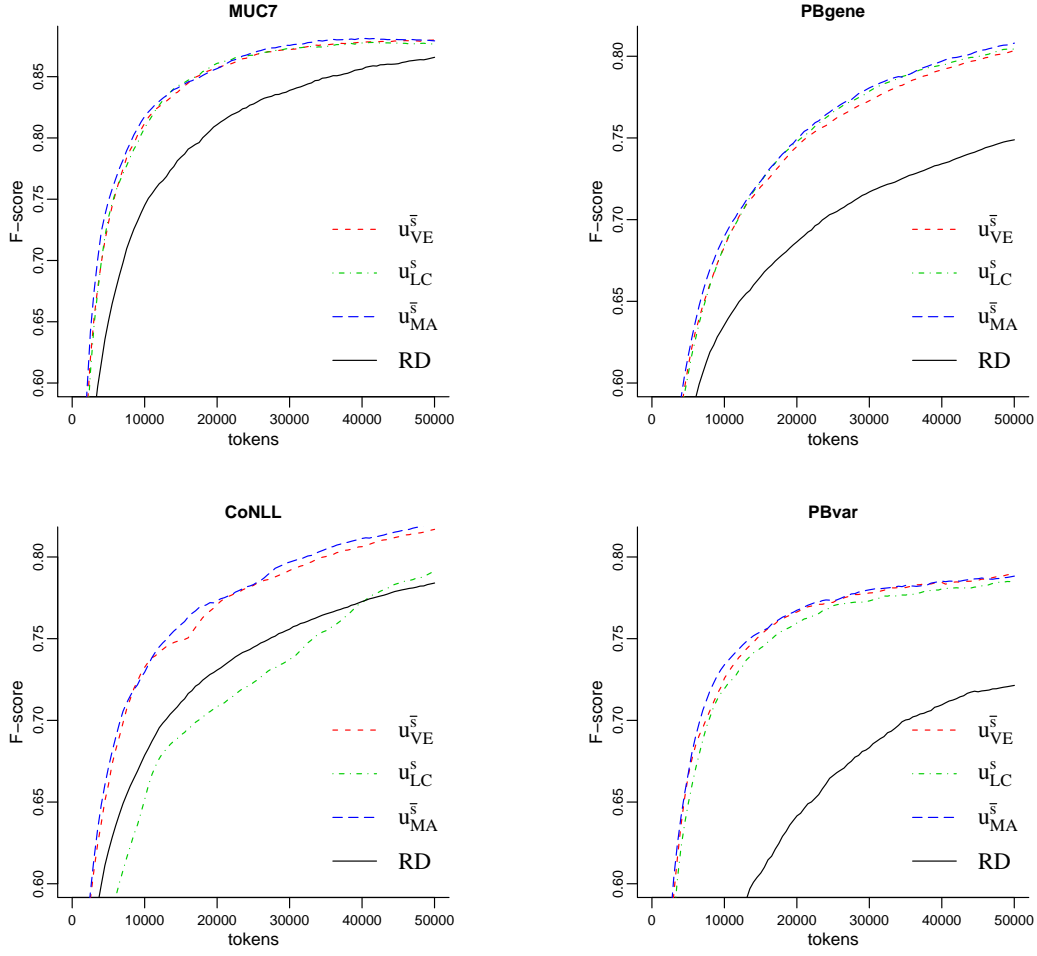


Figure 4.3: Learning curves for selected utility functions and random selection.

do only slightly worse.⁹ The percentage decrease of annotation effort

$$\Delta\text{CFP}_{F^*}(u_{\text{MA}}^{\bar{s}}, \text{RD}) = 1 - \frac{\text{CFP}_{F^*}(u_{\text{MA}}^{\bar{s}})}{\text{CFP}_{F^*}(\text{RD})} \quad (4.15)$$

of $u_{\text{MA}}^{\bar{s}}$ compared to RD is shown in Table 4.6 which emphasizes the immense potential of AL for reducing annotation effort.

⁹Once again, note the exception on the CoNLL corpus where u_{LC}^s performs even worse than random selection while the other two metrics do well on this corpus. This behaviour is explained by some sample characteristics which are problematic on CoNLL (discussed in detail below).

	MUC7	CoNLL	PBGENE	PBVAR
$\Delta\text{CFP}_{F^*}(u_{\text{MA}}^{\bar{s}}, \text{RD})$	49.3%	47.8%	61.7%	81.8%

Table 4.6: Percentage reduction of sampling complexity through AL based on the $u_{\text{MA}}^{\bar{s}}$ utility function over random selection. Table 4.4 on page 65 shows the respective CFP scores.

4.4.2.2 Characteristics of Selected Samples

This section analyzes characteristics of the samples obtained by AL with different utility functions. The samples analyzed amount to 20,000 tokens.¹⁰ Besides RD, two additional baselines were used. LNG selects sentences by their sentence length so that *longer* sentences are preferred; SRT works inversely and prefers *shorter* sentences. Both SRT and LNG are inferior to any of the AL approaches both in terms of sampling complexity and sampling efficiency.¹¹ Samples obtained by RD are considered to mirror the true distribution of data and the unbiased characteristics of the respective corpus to which then all other metrics are compared. Table 4.7 shows sample characteristics on all four corpora.

Sentence Length (SL) The average sentence length in terms of tokens.

The longest sentences are selected by utility functions based on sequence confidence including u_{LC}^s and u_{MA}^s . Due to the way sequence confidence is calculated (cf. Equation 2.24 on page 19), there is an inherent tendency for longer sentences to obtain higher utility scores and thus to be selected preferentially. This is evidenced by the SL scores. In a similar manner, u_{FC}^s also selects longer sentences than RD.

The opposite is the case for utility functions where the mean average aggregation is applied: Shorter sentences are more likely to obtain a high utility score because local peaks, i.e., high utility scores on single tokens, have greater effect on the averaged utility score.

While the sentence length characteristic described can be observed on all corpora, it is most pronounced on CoNLL where u_{LC}^s exhibits SL scores almost twice as high as RD. This is due to CoNLL’s specific distribution of sentence

¹⁰The same tendencies can also be found in a less extreme version after 50,000 tokens. The effects are less extreme because in later AL iterations, the selection is much more restricted due to a diminishing pool. Thus, to study selection characteristics of AL it makes sense to consider fairly iterations of the AL process. We could observe the same tendencies over all corpora.

¹¹For example, the sampling complexity on MUC7 for the target performance of $F=0.85$ is 37,518 tokens for RD, 36,013 tokens for LNG and 35,512 tokens for SRT. AL with the worst performing utility function needed only 26,014 tokens.

lengths. Moreover, the poor performance of u_{LC}^s results largely from the poor relation between costs and true benefit in model induction from long sentences.

Entities per Sentence (EPS) The average number of entity mentions per sentence.

Statistically, the chance of finding many entity mentions is higher in long sentences compared to short sentences. Accordingly, the highest EPS scores are yielded by u_{LC}^s and u_{MA}^s . Yet, the EPS scores of the other utility functions are higher compared to RD. SLG also exhibits a high EPS scores. However, the selection of sentences with many entity mentions alone cannot explain the good performance of the AL approaches which is emphasized by the poor performance of SLG.

Entity Length (EL) The average entity length in terms of tokens.

There is an overall tendency for AL to select longer entity mentions.

Percentage Entity Tokens (PET) The percentage of tokens that is part of an entity mention – higher scores mean higher density of entity mentions per sentence.

To start with, RD shows that the number of entity tokens is very low in all corpora (between 3% and 14%). All utility functions, except u_{FC}^s , yield considerably higher PET scores. The behaviour of u_{FC}^s might be explained by the following hypothesis: the more short entity mentions contained in a selected sentence, the higher are the chances that the mutual F-scores will differ more. This assumption is supported by the fact that the EL score for u_{FC}^s is also a bit lower compared to the other utility metrics.

In summary, when the number of tokens that are part of entity mentions is considered to be the interesting part of a sentence, we may assert that the use of AL considerably increases the information density.

Percentage Entity Sentences (PES) The percentage of sentences containing entity mentions — higher scores mean that more sentences contained entity mentions.

As with PET, the number of sentences containing entity mentions is considerably increased by selective sampling. The sequence confidence-based utility functions exhibit the highest PES scores again because long sentences are statistically more likely to contain entity mentions.

The biggest reduction of annotation effort (cf. Table 4.6 on page 68) was achieved on the PBVAR corpus can be explained by the low PET score of 0.23 on this corpus – statistically, less than one out of four sentences contains an entity mention. This can be further intensified by the fact that entity mentions often co-occur.

The above analysis demonstrates that different utility functions lead to specific sample characteristics. Most notably, sentences selected by any of the utility functions

except u_{FC}^s exhibit a higher information density, which we loosely defined as percentage of tokens covered by an entity mention (PET). Moreover, AL tends to select either overly long or short sentences, based on the specific utility function used. However, sentence length alone cannot explain the good performance of AL, as LNG and SRT perform rather poor in comparison. Overall it can be presumed that AL is an efficient sampling approach especially when the pool of unlabeled examples is sparsely populated with entity mentions so that a sampling bias towards those sentences containing many entity mentions is very beneficial.

4.4.2.3 Evaluation of the Selection Granularity

In Tomanek et al. (2009), we showed that the sentence-level selection granularity increases the robustness of AL compared to the more fine-grained token-level selection. The sampling bias induced by AL may lead to an incomplete coverage of the input space which has been described as the missed cluster effect by Schütze et al. (2006): Larger parts, or clusters, of the input space are completely missed during the AL selection process. AL is an exploitative sampling process. Starting from a seed set that is limited to a certain subspace of the input space, AL may not be able to explore other regions of the input space. Explorative approaches to sampling, such as random sampling, tend to cover the complete input space more uniformly.

By using the sentence-level selection granularity, non-targeted parts of the input space may be also covered by a sentence that was selected because another part of the same sentence appeared highly useful for classifier training. In consequence, AL based on sentence selection recovers better from unfavourable seed sets. We have described this behavior as the *co-selection* effect (Tomanek et al., 2009). Altogether, this also means that AL with a sentence-level granularity also constitutes a combination of exploitative and explorative sampling processes.

4.5 Summary and Conclusions

In summary, this chapter discusses the NLP task of NER and presents and evaluates our NER-specific AL framework. This framework is an instantiation and adaptation of the general framework of greedy AL defined, in a task-independent manner in the previous chapter. This framework is used for all experiments in this thesis.

We first described our approach to NER based CRFs and a rich set of domain-independent features. The evaluation on four standard corpora annotated with NEs showed that – although not specialized to any of the specific NE tasks tested on – this approach yields performance values not much below those of highly tuned

MUC7 corpus						CoNLL corpus					
strat.	SL	EPS	EL	PET	PES	strat.	SL	EPS	EL	PET	PES
RD	25.8	1.7	1.6	0.1	0.7	RD	15.7	1.5	1.5	0.2	0.7
SRT	15.4	0.9	1.5	0.1	0.6	SRT	5.3	1.0	1.4	0.3	0.7
LNG	48.5	3.3	1.6	0.1	0.9	LNG	41.6	2.8	1.6	0.1	0.9
$u_{VE}^{\bar{s}}$	24.5	2.6	1.8	0.2	0.9	$u_{VE}^{\bar{s}}$	9.8	1.6	1.6	0.3	0.8
u_{FC}^s	25.1	1.6	1.7	0.1	0.9	u_{FC}^s	16.6	1.3	1.4	0.1	0.8
$u_{KLM}^{\bar{s}}$	25.4	2.8	1.8	0.2	0.9	$u_{KLM}^{\bar{s}}$	10.6	1.8	1.7	0.3	0.9
u_{LC}^s	33.8	3.4	1.7	0.2	0.9	u_{LC}^s	28.5	3.7	1.6	0.2	1.0
u_{MA}^s	32.2	3.1	1.7	0.2	0.9	u_{MA}^s	24.8	3.0	1.5	0.2	0.9
$u_{LC}^{\bar{s}}$	25.2	2.7	1.7	0.2	0.9	$u_{LC}^{\bar{s}}$	10.5	1.8	1.6	0.3	0.9
$u_{ENT}^{\bar{s}}$	24.8	2.7	1.7	0.2	0.9	$u_{ENT}^{\bar{s}}$	10.2	1.9	1.6	0.3	0.8
$u_{MA}^{\bar{s}}$	25.3	2.7	1.7	0.2	0.9	$u_{MA}^{\bar{s}}$	10.4	1.8	1.6	0.3	0.9

PBGENE corpus						PBVAR corpus					
strat.	SL	EPS	EL	PET	PES	strat.	SL	EPS	EL	PET	PES
RD	25.3	1.2	1.3	0.1	0.7	RD	25.3	0.6	1.5	0.03	0.2
SRT	12.1	0.7	1.2	0.1	0.5	SRT	12.0	0.2	1.5	0.03	0.1
LNG	52.8	2.4	1.4	0.1	0.8	LNG	52.7	1.5	1.4	0.04	0.4
$u_{VE}^{\bar{s}}$	22.6	2.3	1.8	0.2	0.9	$u_{VE}^{\bar{s}}$	22.8	2.3	1.5	0.2	0.7
u_{FC}^s	26.9	1.6	1.7	0.1	0.9	u_{FC}^s	28.7	1.6	1.4	0.1	0.6
$u_{KLM}^{\bar{s}}$	23.7	2.3	1.9	0.2	0.9	$u_{KLM}^{\bar{s}}$	23.9	2.5	1.5	0.2	0.7
u_{LC}^s	37.8	3.6	1.5	0.1	0.9	u_{LC}^s	36.4	3.5	1.4	0.1	0.7
u_{MA}^s	33.6	3.0	1.5	0.1	0.9	u_{MA}^s	32.9	2.8	1.4	0.1	0.7
$u_{LC}^{\bar{s}}$	23.6	2.4	1.8	0.2	0.9	$u_{LC}^{\bar{s}}$	24.0	2.5	1.5	0.2	0.7
$u_{ENT}^{\bar{s}}$	24.0	2.5	1.7	0.2	0.9	$u_{ENT}^{\bar{s}}$	24.8	2.6	1.5	0.2	0.7
$u_{MA}^{\bar{s}}$	23.6	2.4	1.8	0.2	0.9	$u_{MA}^{\bar{s}}$	23.8	2.4	1.5	0.2	0.7

Table 4.7: Characteristics of samples obtained by different sampling strategies including random sampling (RD), shortest sentence selection (SRT), longest sentences selection (LNG), and AL based on different utility functions (u_{NAME}). Characteristics were calculated as soon as the samples yielded a size of 20,000 tokens.

systems. This positive results justify CRFs as selector and consumer in all of the following AL experiments in the context of NER.

A subset of the general AL utility functions described in Chapter 3 was adapted to work in the NER-specific AL framework. The performance of all of these utility functions was comprehensively evaluated and we identified three utility functions that outperformed the others and will thus be used in the following chapters of this thesis. Note that while for the QbC-based utility functions we found big differences in the sampling performances, this was not the case for the US-based ones where there was minor variance in performance. Furthermore, we could not find a fundamental difference in sampling complexity or sampling efficiency between the QbC or the US-based approaches in general.

However, taking into account the slight performance differences, $u_{MA}^{\bar{s}}$ showed the best performance values. Compared to random sampling, we recorded a reduction of annotation effort through AL with the $u_{MA}^{\bar{s}}$ utility function of between 47.8 and 81.8 % (cf. Table 4.6 on page 68). These savings still differ from the exponential savings AL can yield in theory and under optimal conditions (cf. Section 3.4). However, in practise those savings are motivating for AL to be considered an efficient strategy for considerably reducing the burden of annotation.

AL always performed better than random sampling – with one exception: On the CONLL corpus, u_{LC}^s showed extremely poor performance in terms of sampling efficiency measured by the RAI score. Our analysis of the sample characteristics of the different utility functions could provide an explanation for that. Sequence confidence-based utility functions tend to select overly long sentences, utility functions based on an aggregation of token-level utility scores rather select shorter sentences. On the CONLL corpus, sentence length is subject to high standard deviation and long sentences exhibit linguistic structures that are not found in shorter sentence and that do not help much in model learning. In consequence, preferential selection of long sentences is extremely disadvantageous on this corpus. On the other corpora, however, selection of extremely long sentences did not constitute a problem.

Another sample characteristic common to all utility functions is the high information density, i.e., an increased number of tokens part of entity mentions in the selected sentences. From this, we conclude that AL is especially beneficial in scenarios where this information density is naturally rather low – as is the case in NER.

Finally, our NER-specific AL framework combined with the three selected utility functions, $u_{VE}^{\bar{s}}$, u_{LC}^s , and $u_{MA}^{\bar{s}}$, satisfactorily meets the criteria of Requirements 1, 2, and 3: the selection of the next batch of sentences to be annotated exhibits a relatively low complexity, the framework can be used flexibly for different NER scenarios, and lastly, relevant savings can indeed be achieved.

Chapter 5

Monitoring the Active Learning Sampling Process

When AL is applied to real-world annotation endeavours, a crucial question is when to actually stop the annotation process and cash in the savings in annotation effort (Requirement 4). This question may be asked both in annotation projects based on AL and in those where traditional strategies, such as random sampling, are applied to sample the raw corpus data. However, this question naturally comes up in the context of resource- and cost-aware annotation strategies of which AL is one.

Performance gains in classifier training are usually sub-linear, i.e., gains rapidly slow down with each new training example available.¹ A learning curve can be subdivided into three stages – a short skyrocket stage at the beginning with an extremely steep slope, a transition stage with mediate slope, and a long saturation or convergence stage. These stages are shown on a prototypical learning curve in Figure 5.1. This plot also visualizes why it is important to find a proper stopping point. Stopping too early means that precious gains in classifier performance at low cost are forfeited (S1); stopping too late leads to human annotation effort being wasted (S4). The optimal stopping point is probably between both extremes, in the transition or at the beginning of the saturation stage (S2 and S3).

What is the *optimal* stopping point, though? Stopping once a user-defined target performance has been reached may be a questionable stopping condition because it cannot be guaranteed that the target performance can be reached at all, leading to an infinite annotation process in the worst case. Annotation should be stopped at the latest when the best-performing classifier for a particular problem has been yielded on the data at hand, so that further annotation will no longer improve the model. In most real-world annotation scenarios, however, a well-defined stopping point based on the convergence of classifier performance does not exist. Instead,

¹The relationship between number of training data and classifier performance obtained hereon is a polynomial one known as the power law (John and Langley, 1996).

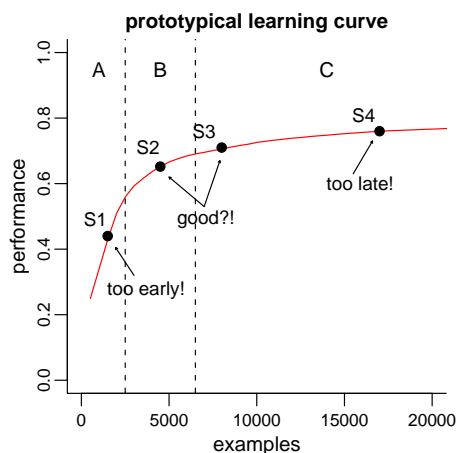


Figure 5.1: Prototypical learning curve subdivided into three stages: the start-up stage (A), the transition stage (B), and the saturation stage (C). $S1 - S4$ are possible stopping points within these stages.

additional data still result in slight improvements of the classifier’s performance. Accordingly, one might rather consider the *trade-off* between further annotation efforts and gains in classifier performance to decide whether additional annotations are worth the effort for the targeted application. In consequence, we consider the *optimal* stopping point to be subjective.

Given a user-defined trade-off, the stopping point can be read from the learning curve. Unfortunately, a learning curve is rarely available in practise. Classifier performance cannot be reliably estimated by means of re-sampling methods, such as cross-validation or bootstrapping, because these methods require *i.i.d.* examples (cf. Section 2.3). Samples drawn with AL are, however, intentionally biased so that an annotated validation set with *i.i.d.* examples is needed for reliable performance estimation. Yet this solution comes with expensive extra annotation work, which is not consistent with the goal of AL of keeping annotation effort to a minimum. Also, if there were a sufficiently large validation set, AL would not be needed.

In this chapter we propose an unsupervised method for approximating the learning curve without the need for an *annotated* validation set. The approximation shows the general progression of the learning curve and makes it possible to identify the current stage of the learning curve. Our method is basically a means of monitoring the annotation progress. Given our observation that the actual stopping point depends on a subjective trade-off definition, the approximation of the learning curve is an important aid to decision making. As an extension to the monitoring approach, we also propose an intrinsic stopping criterion for finding a reasonable stopping

point close to the best-performing model. This stopping condition may be used in scenarios where the definition of a trade-off is not possible and stopping should thus be done in a conservative way.

Most of the work presented in this chapter has been published in Tomanek et al. (2007a), Tomanek and Hahn (2008), and Olsson and Tomanek (2009).

5.1 Related Work

Early stopping is a general issue in the machine learning literature. However, the setting in which it is defined differs considerably from ours. A challenging problem of data mining is the enormous amount of data collected. Sampling techniques are applied to reduce computational complexity in classifier training – be it for main memory restrictions or reduction in training time. Determining the stopping point is about finding an appropriate trade-off between computational complexity in model induction and reduction of model performance (Scholz, 2007, Ch. 3).

In this context, extrapolation of the learning curve from the history of performance measurements on small to medium sized samples has been explored. John and Langley (1996) employed a power law function to fit learning curve data and predict classifier performance for future sample sizes.² It should, however, be noted that the AL scenario is different in that a history of performance measurements is not available in practise (see discussion in previous section). The approach we propose for monitoring and stopping is, however, related to this work as it aims to approximate a learning curve, though not from performance measurements but rather from utility scored from the AL process.

Schohn and Cohn (2000) proposed a stopping criterion for SVM-based AL where the annotation process is stopped when none of the unlabeled examples are closer to the hyperplane than any of the support vectors. At this point, the margin has been exhausted and the model will no longer change. This approach assumes that the best stopping point is that of model convergence. In this way, it is a rather conservative stopping condition and it is questionable whether it can be reached in a practical scenario with pools of virtually unlimited size.

Several stopping criteria for US-based AL have been proposed by Zhu and colleagues (Zhu and Hovy, 2007; Zhu et al., 2008a,b). The max-confidence method stops AL

² Based on theoretical work in both machine and human learning it has been shown that the power law is a good fit to the respective learning curves (Ninio, 2006). The power law function is given by $acc \approx a - b \cdot n^\alpha$ where a , b , and α are estimated from previous performance measurements and n denotes the size of the sample.

when the utility score on any of the selected examples falls below a certain threshold. Similarly, the overall-confidence method stops AL when the average utility score on the unlabeled examples in the pool falls below a threshold. Using the min-error strategy, AL is stopped when the difference between the classifier’s predictions and the labels given by the human annotator for the selected examples falls below a threshold. The minimum expected error method estimates the classification error on the remaining pool of unlabeled examples. Once again, when this error falls below a threshold, AL is stopped. Finally, the classification-change method stops AL when, during two consecutive AL iterations, the predicted labels on unlabeled examples in the pool do not differ. Altogether, all methods proposed by Zhu and colleagues are based on the AL pool alone. We will discuss below why this is highly problematic. Moreover, all of the methods require a threshold to be set.

Vlachos (2008) presented a stopping criterion for US-based AL based on the confidence of the model learned in the current AL iteration. This confidence is estimated as the average utility score (here, least confidence) of the model on a held-out test set. Vlachos argued that the optimal stopping point is the point when there are no more useful examples left in the pool. Thus, as soon as the confidence values stop increasing, this point is reached. Vlachos reported that such a confidence curve follows a rise-peak-drop pattern: It rises at the beginning, then reaches its maximum values after which it then constantly drops.

Vlachos’ approach was re-evaluated by Laws and Schütze (2008), but they could not find the peak-pattern of the confidence curve in their setting. Laws and Schütze (2008) proposed a related stopping criterion also based on the convergence of the utility score in a US scenario. They observed the gradient of the utility score of the last selected example, i.e., the most useful example. Since utility scores are often noisy and subject to sharp drops, the gradient is calculated from a moving median over the utility scores. When the gradient approaches a value of 0, the AL process is stopped due to the intuition is that at this point the pool of available data no longer contributes to the classifier’s performance.

Most recently, Bloodgood and Shanker (2009a) proposed a stopping criterion based on stabilizing predictions. Predictions on a so-called stop set of the models learned in n subsequent AL iterations are compared using Cohen’s Kappa statistic κ (Cohen, 1960). When κ is above a threshold, the predictions, as well as the underlying model, are assumed to be stable justifying to stop annotation. Bloodgood and Shanker argue that their criterion is robust and widely applicable because the Kappa statistic is more robust than simple percent agreement calculation. Kappa takes into account the agreement occurring by chance. In their experiments on text classification and NER, Bloodgood and Shanker successfully applied the same threshold of $\kappa = 0.9$. Moreover, the authors argued that their stopping criterion is aggressive as it avoids

unnecessary annotations. They empirically compared previous stopping criteria and find that their own criterion was the most aggressive and stable one, leading mostly to the best stopping decision.

Acknowledging that the best stopping point depends on a subjective trade-off between annotation effort and performance improvement – an issue not picked up by the other studies on stopping – Bloodgood and Shanker argued that the aggressiveness of their stopping criterion can be controlled by the Kappa threshold. However, the Kappa threshold is not directly linked to a particular trade-off scenario, so that it is questionable whether this adjustability is of practical advantage.

5.2 Monitoring and Stopping Committee-based AL

Our approach to monitoring and stopping the AL process was developed and published at a time when other groups were also working on the same issue. While our approach to stopping was independently developed, there are still overlaps with other works. In the following section, our approach is described in detail and differences as well as similarities to other approaches are underlined.

Put briefly, the most important differences of our approach are that it is developed for committee-based AL (all other approaches were formulated for US-based AL) and that the use of a separate held-out test set is an integral part (most other approaches rely instead on the pool). A commonality of all sophisticated approaches to stopping, including ours, is that they are based on the utility scores being the fundament of any AL approach.

5.2.1 Approximating Learning Curves

When a learning curve is available, stopping points can easily be identified by means of a particular trade-off between annotation costs and performance gains. Due to the absence of learning curves in real-world annotation scenarios, other means for monitoring and stopping the AL process are needed. However, based on the insight that the optimal stopping point is but a subjective one determined by user- or application-specific trade-off definitions between costs and gains, we tried to find a way to approximate the learning curve, to which the trade-off could be applied.

Based on the insight that learning curves are subject to an asymptotically converging shape our approach aims at approximating this shape without the need for actual performance measurements. When we plotted both the learning curve of an AL process and AL-specific utility scores over time, we observed that classifier performance

increased and utility scores decreased in approximately the same manner. Based on this observation, we hypothesized that the progression of the learning curve could be approximated by the progression of the utility scores. We developed our approach in the context of committee-based AL and the Vote Entropy utility function. $u_{\text{VE}}^{\bar{s}}$ is based on divergent predictions by the committee members. When the committee members no longer disagree, classifier performance will probably have stopped increasing. Also, at the time when the committee exhibits high $u_{\text{VE}}^{\bar{s}}$ scores, the generalization performance of each single committee member is not yet good, so that further data can be expected to cause high performance gains.

However, on which data should we calculate the $u_{\text{VE}}^{\bar{s}}$ scores used to approximate the learning curve? During the AL process itself, $u_{\text{VE}}^{\bar{s}}$ scores are calculated on all unlabeled examples anyway. In the most straightforward approach, one might simply calculate the average of the $u_{\text{VE}}^{\bar{s}}$ scores over the batch \mathcal{B} of selected examples which we call the *selection agreement* (SA):

$$\text{SA} = 1 - \frac{1}{|\mathcal{B}|} \sum_{p \in \mathcal{B}} u_{\text{VE}}^{\bar{s}}(p) \quad (5.1)$$

The intuition is that the committee will agree increasingly on the hard examples selected from the diminishing pool as the AL process proceeds. When the members of the committee are in complete agreement, $\text{SA} \approx 0$, and this is the latest point where AL should be aborted since it will no longer contribute to the overall learning process – in this case, AL is but a computationally expensive counterpart of random sampling.

The SA, however, is affected by the diminishing size of the pool of unlabeled examples. In simulation settings, the pool is of a very limited size – normally only a few thousands of examples. As a consequence, the total number of positive and hard examples, which are preferentially selected by AL, is rather limited. Our experiments in the NER scenario showed that sentences containing many and complex entity mentions are selected in early AL iterations, so that in late AL iterations hardly any useful examples are left in the pool. As a consequence, it is only in early iterations that AL really has the choice to select useful examples and the SA naturally mounts to values close to 1 in later iterations.

The SA curve profits from this *simulation artifact*. At the time when no more interesting examples are left in the pool, it hits the 100% agreement line. At the latest at this point AL should be stopped and this is the position where the learning curve (in simulations!) does not rise any more. In real-world annotation scenarios where the pool is of virtually unlimited size and much more diverse, there will always be useful (and thus difficult) examples that AL may find, thus keeping the

selection agreement constantly high. In consequence, it cannot be used as a reliable approximation of the learning curve for real-world settings and its functionality is a essentially an artifact of the simulation scenario.

In fact, we claim that all monitoring and stopping approaches based on utility scores measured on the diminishing pool are subject to this simulation artifact which renders them questionable in real-world settings. The simulation artifact may be specifically problematic for the min-error strategy proposed by Zhu et al. (2008b). The reason why min-error works very well in simulations is presumably that after some AL iterations all hard examples are already annotated, so naturally the accuracy goes up as predictions are made on simple examples, only. Given an extremely large and diverse pool, AL will always find critical examples, so that min-error will presumably not work as a stopping criterion.

As a solution to the simulation artifact, we propose to calculate the average agreement for each AL iteration on a separate, held-out validation set \mathcal{V} . The *validation set agreement* (VSA) score is thus defined by

$$\text{VSA} = 1 - \frac{1}{|\mathcal{V}|} \sum_{p \in \mathcal{V}} u_{\text{VE}}^{\bar{s}}(p). \quad (5.2)$$

\mathcal{V} should reflect the real data distribution and must not be used in the annotation process itself. For most NLP tasks, unlabeled data is virtually unlimited. The validation set comes at no extra costs as no annotations are required on it. Calculating VSA scores poses minimal extra computational complexity; the committee, trained during each AL iteration anyways, only has to be applied to \mathcal{V} .

Plotted over time we get the VSA curve. Since the validation set stays the same throughout the entire AL process, VSA values of consecutive AL iterations are comparable. Moreover, since the examples of the validation set are not used in the annotation process, the VSA is only affected by the performance of the committee, which, in turn, is grounded in the information contained in the most useful examples selected from the pool of unlabeled data.

Now, from a VSA curve that ascends only slightly between selected measurement points we can infer that the respective learning curve has only a low slope at these positions, too. Although it is not possible to interpret the actual agreement values of the VSA curve, its progression behavior can be used to estimate whether further annotation is worth the human labeling effort. Below, we provide empirical evidence that the VSA curve is indeed an adequate approximation of the learning curve progression and that the SA curve fails in the real-world annotation scenario where examples are selected from a much larger pool.

This approximation of the learning curve is new and unique in the following ways: based on the acknowledgement that stopping points are subject to user-specific preference structures, we have created a means to apply personal trade-off definitions to an approximated learning curve. While the approximation does not allow exact trade-offs to be applied, one can still identify at which stage (start-up, transition, or saturation) the learning process is. In doing this, the only parameter a user has to provide is a trade-off. If no such trade-off is available, one could also just monitor the VSA curve and, based on its progression, decide whether to continue annotation. When the VSA curve exhibits an extremely low slope, this is a hint that the learning curve will also not increase much more, so that annotation may be stopped.

This approach, published in Tomanek and Hahn (2008), is very similar to the one recently suggested by Bloodgood and Shanker (2009a). Both approaches work on a separate validation set and are based on some kind of stabilizing predictions. In the case of Bloodgood and Shanker’s approach, stabilization of predictions is directly observed, while in our approach it is indirectly observed through the $u_{VE}^{\bar{s}}$ utility score. The main difference of our approach is that we leave the final stopping decision to the user but offer decision support by means of a monitoring tool. Bloodgood and Shanker, in contrast, require the user to specify a threshold for stopping. Setting this threshold to a reasonable value is difficult because there is no general relation between the actual threshold value and classifier performance or trade-off between model performance gains and annotation costs.

5.2.2 Objectivizing Stopping

While we claim that the VSA curve can be used to approximate the progression of the learning curve, to which a user-defined trade-off can be applied to find a stopping point, we propose an extension to this approach, where an objective stopping point is defined. Such a stopping point may be valuable when the user has no concrete conception of a proper trade-off. The proposed stopping point is an alternative which stops at a reasonable point based on learning and distributional aspects. Since this stopping criterion does not require any external parameters to be specified and relies on characteristics of the data and the learning algorithms only, we call it *intrinsic*. This approach has been published in Olsson and Tomanek (2009).

In early AL iterations, the SA is usually lower than the VSA. The *intrinsic stopping criterion* (ISC) is defined as the point where the SA curve and the VSA curve cross with the result that in the current AL iteration

$$SA > VSA. \tag{5.3}$$

This crossing marks the point when the pool of unlabeled examples has been exhausted to a certain extent so that SA is low and the performance of the model learned is close to the maximal performance that can be yielded on the given pool. When the SA exceeds the VSA, this means that the committee is more in agreement concerning the most useful examples in the (diminishing) pool than it is concerning the held-out validation set. From this we can infer that the committee would learn more from a random sample from the validation set (or from a data source exhibiting the same distribution of examples), than it would from the unlabeled data pool.

This tells us that the pool has been considerably harvested so that its informative content has shrunk. In consequence, low gains in classifier performance by additional annotation can be expected, so this marks a reasonable point to stop if no other information about the user’s stopping preferences is available. In this way, the ISC is rather conservative, which may, however, be a good alternative in the absence of a user-defined trade-off.

A major advantage of the ISC is that it relies only on the characteristics of the selector and the data at hand and so does not require the user to set any external parameters prior to initiating the AL process. Furthermore, the ISC is designed for QbC-based AL and as such it is independent of how disagreement between the committee members is quantified.

5.3 Experiments

5.3.1 Experimental Settings

We ran several experiments on the simulation corpora in the same settings as in Chapter 4. u_{VE}^s was applied as utility function in the committee-based AL setting. As a held-out set to calculate the VSA values we employed the test sets of the respective corpus which is also used to generate the learning curve. For calculation of the VSA values, the label information contained in the test set was ignored. Reported results are averages over five independent runs. The single runs were long-term runs, i.e., they were continued until the complete corpus or 150,000 tokens had been selected so that convergence of the learning curve could be recorded.

To test the robustness of our approach to monitoring and stopping AL, we also applied it in two real-world AL annotation initiatives performed in the context of a biomedical research project. This included, (a) the CYTOREC project, where entity mentions of cytokine and growth factor receptors had to be annotated, and (b) the CDANTIGEN project, which focused on the annotation of entity mentions of

immunologically relevant antigens.³ For both annotation projects, the pool consisted of approximately two million sentences taken from PUBMED abstracts. A test set of 2,165 sentences was annotated for the purpose of evaluating AL in terms of a learning curve. Moreover, the VSA curve was also calculated from this set, ignoring the labels. AL was started from non-random seed sets which were generated so as to contain many relevant entity mentions from the outset.⁴

5.3.2 Results

Monitoring the Learning Progress Figures 5.2 and 5.3 display the learning and agreement curves on the four simulation corpora. Learning curves are shown so that the *approximated* progression of the learning curve, obtained from the agreement curves, can be compared to its *true* progression. As for the agreement curves, exact agreement values (dots) and a curve obtained by local polynomial regression fitting (solid line) are shown.⁵

On all corpora, the SA curve reaches a point of convergence. When this point is reached, the learning curve too has stagnated. However, the VSA curve describes the progression of the learning curve better, especially in the transition stage. Consider, for instance, the CoNLL corpus where the SA curve has a relatively steep, but almost constant slope between about 35,000 and the convergence point (about 115,000 tokens). This can also be observed on the PBGENE corpus and in a less pronounced manner also on the MUC7 corpus. On the VSA curve, in contrast, the slope changes and becomes increasingly smaller until it (almost) converges. Moreover, this (quasi-) convergence stage is reached earlier than on the SA curve.

Figure 5.4 displays the learning and agreement curves for the real-world annotation projects.⁶ The learning and agreement curves start at 10,000 (CYTOREC) and 40,000 (CDantigen), in order not to evaluate a learning curve on the fairly large, non-random seed sets. On the CDANTIGEN corpus, after 80,000 tokens have been annotated the learning curve has not completely converged. The AL-based annotation process was stopped here because additional annotations would not pay off very much. The VSA curve mirrors this behavior and continues to ascend with a

³For more details see Chapter 12.

⁴The heuristic to generate the seeds is described in detail in Tomanek et al. (2007b).

⁵Local polynomial regression fitting as implemented in the `loess` function provided by R is applied with default parameters (Team, 2008).

⁶During the actual AL-based annotation process, no VSA scores were calculated. Instead, these were calculated in an ex-post setting for this evaluation. Due to the randomness when sampling the committee, we averaged over three runs where we calculated the agreement curves and took the agreement scores after every fifth AL iteration.

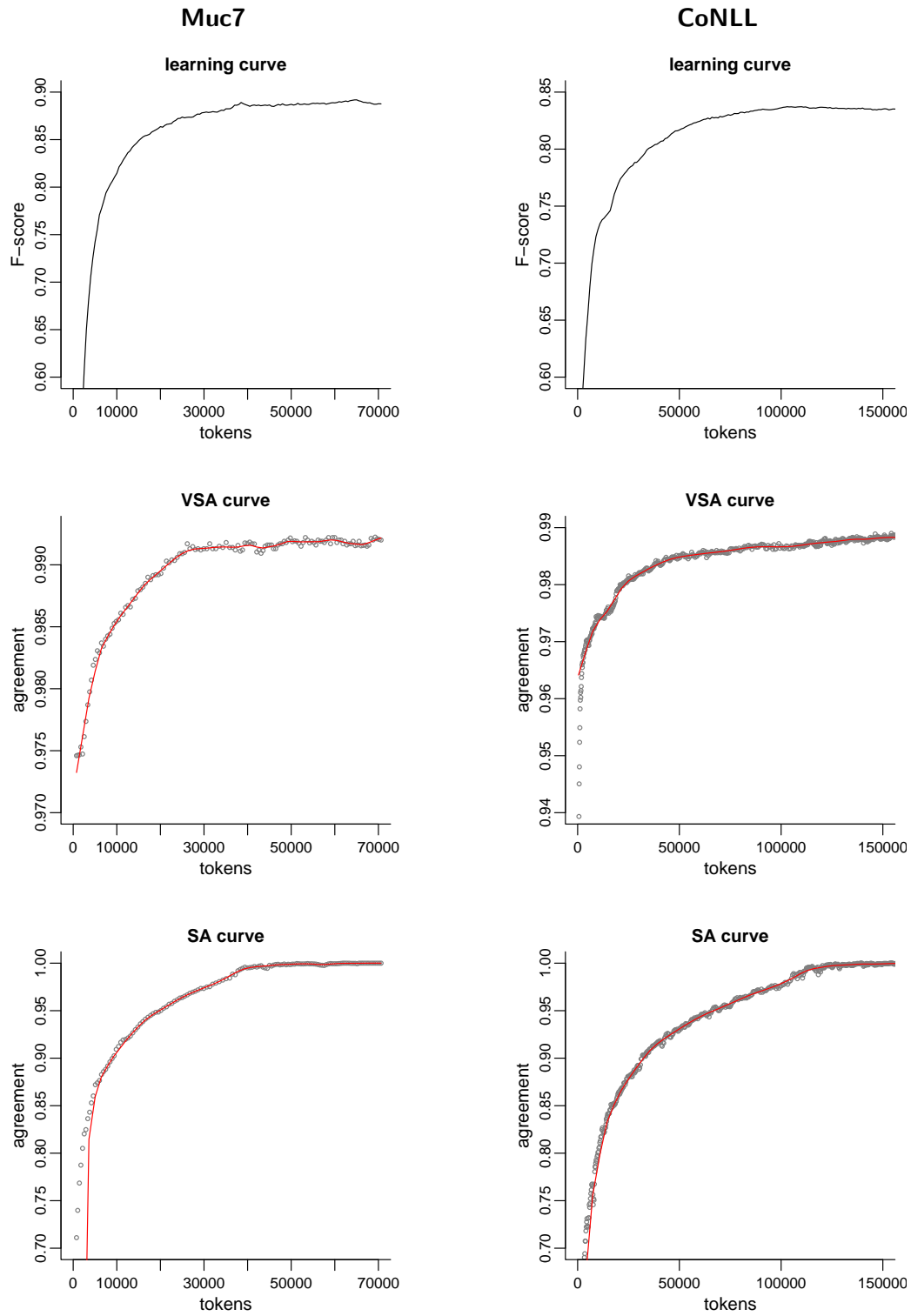


Figure 5.2: Learning and agreement curves for simulation corpora.

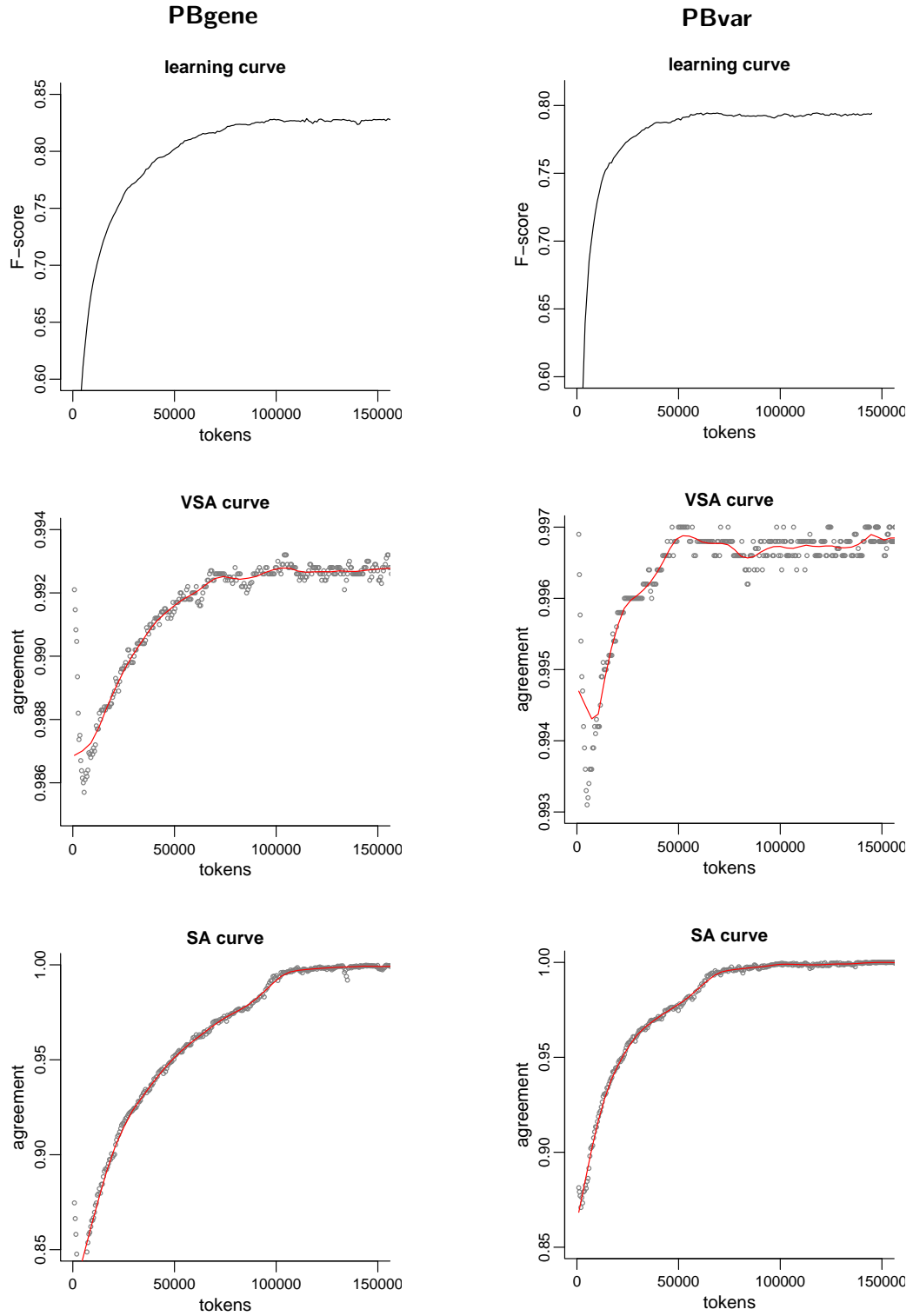


Figure 5.3: Learning and agreement curves for simulation corpora.

very low slope. The SA curve, in contrast, remains quite obscure and is subject to extreme deviations.

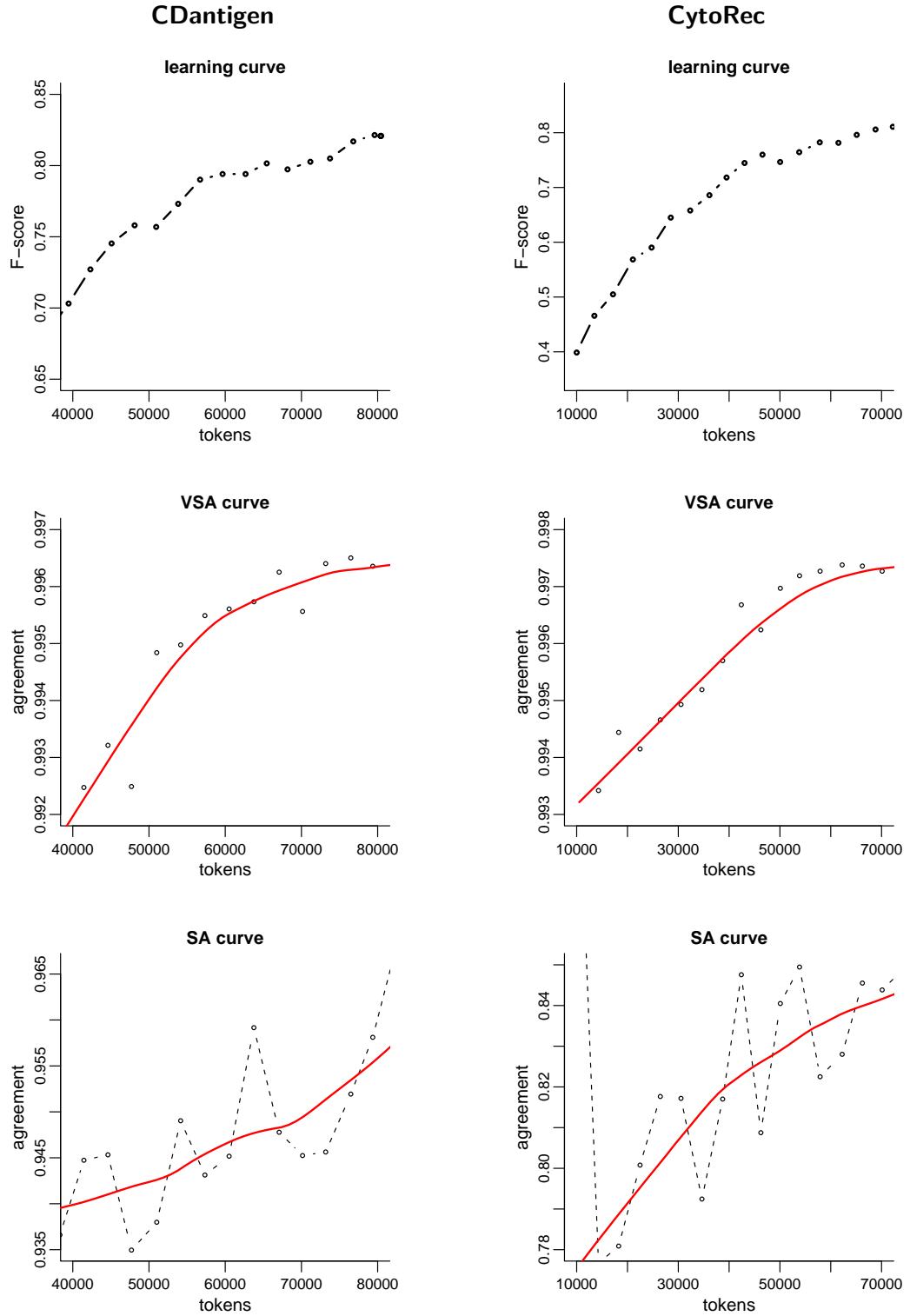
A similar behavior can be observed for the CYTOREC corpus. The learning curve is only slightly ascending after about 65,000 tokens have been annotated. Once again, this is mirrored by the VSA curve and the SA curve is hard to interpret. Though its slope decreases somewhat after roughly 40,000 tokens, it continues ascending thereafter. Moreover, both SA curves exhibit an oscillating behavior that contains hardly any clues about the learning curve.

Our experiments showed that in the simulation scenario the two agreement curves (SA and VSA) in principle share a similar curve progression which is mainly due to the simulation effect. However, while the point of absolute convergence can be read quite reliably from the SA curve, the VSA curve more comprehensively models the overall behavior of the learning curve especially in the transition stage. In the real-world annotation scenarios, SA curves do not properly model the progression of the learning curve while the VSA curves nicely approximate the learning curves.

Stopping the Annotation Process The progression of the VSA curve is a proper means of monitoring the learning progress. By itself it does not *finalize* any stopping decisions but it can be employed as a *guide* to do so. As discussed above, the overall assessment for balancing the trade-off between annotation costs and expectable quality gains for the learner is left to the annotation supervisor.

We tested two prototypical trade-off scenarios which we then used to find a stopping point by the help of the VSA curve. In the first trade-off scenario (VSA1) we assumed that the focus was on achieving an acceptable performance with low annotation costs, i.e., annotation will be stopped as soon as the learning rate considerably drops. In the second trade-off scenario (VSA2), we assumed the need for a very high-performing classifier even at the cost of extra annotation effort. These two trade-off scenarios can be translated into stopping points, given the approximated learning curves. VSA1 leads to a rather aggressive stopping criterion and VSA2 to a rather conservative one. There are situations where a trade-off cannot be defined, for example, because an informed annotation supervisor is indispensable. In this situation, the intrinsic stopping criterion (ISC) can be applied as a reasonable stopping point when no information about cost and benefit preferences is available.

We compare the VSA1, VSA2 and the ISC with the stabilizing predictions (SP) stopping criterion proposed by Bloodgood and Shanker. Additionally, all these stopping criteria are contrasted with the point where the maximum AL performance (MAX) is yielded on the respective corpus. Table 5.1 gives an overview of the yielded stopping points. In direct comparison of these points to MAX, one can identify the



86 Figure 5.4: Learning and agreement curves for the real-world annotation scenarios.

stopping approach	evaluation criterion	corpora			
		MUC7	CoNLL	PBGENE	PBVAR
VSA1	tokens	20,000	50,000	65,000	25,000
	F-score	0.86	0.82	0.82	0.77
VSA2	tokens	30,000	100,000	120,000	60,000
	F-score	0.88	0.84	0.83	0.79
ISC	tokens	39,250	109,367	192,562	80,785
	F-score	0.89	0.84	0.83	0.79
SP	tokens	28,065	111,392	65,298	36,726
	F-score	0.88	0.84	0.82	0.79
MAX	tokens	65,016	103,008	115,015	118,013
	F-score	0.89	0.84	0.83	0.80

Table 5.1: Stopping points according to different stopping criteria including “manually” applied trade-offs VSA1 and VSA2, our ISC criterion, Bloodgood and Shanker’s SP criterion, and the point where maximum performance is reached. The stopping points are evaluated in terms of sampling complexity (number of tokens needed) and sampling efficiency (F-score reached) after stopping.

loss of classifier gain as well as the saved annotation effort. Figure 5.5 visualizes our stopping points on both the VSA and the learning curves of MUC7 and PBGENE.

On the MUC7 corpus, for instance, only minor improvements of classifier performance can be assumed after 30,000 tokens according to the VSA curve. Given VSA2, 30,000 tokens are a good point to stop. Here, the classifier exhibits a performance of $F=0.88$. More aggressive stopping according to VSA1 and impatience may result in stopping at 20,000 tokens with a performance of $F=0.86$. Compared to MAX, VSA1 reduces the annotation effort to 67 % at a loss of 2 percentage points (pp) of classifier performance. VSA2 still reduces annotation effort by 50 % with a loss of only 1 pp. ISC is even more conservative and in consequence results in (almost) no performance loss, but on the flipside saves only 35 % annotation effort.

We may summarize the following observations: VSA1 and VSA2 are both highly subjective and were placed manually by inspecting the VSA curve in these experiments. They, however, show that the VSA curve is a good approximation to the learning curve in that it allows identification of the start-up and the transition stage. Moreover, given a user has preference information, a stopping point can be identified manually using the VSA as decision help. ISC is a conservative stopping point with the advantage of low losses of classifier performance. In contrast, SP is more aggressive, leading to lower annotation effort than ISC, though accommodated by higher losses of classifier performance. However, SP is less aggressive than VSA1.

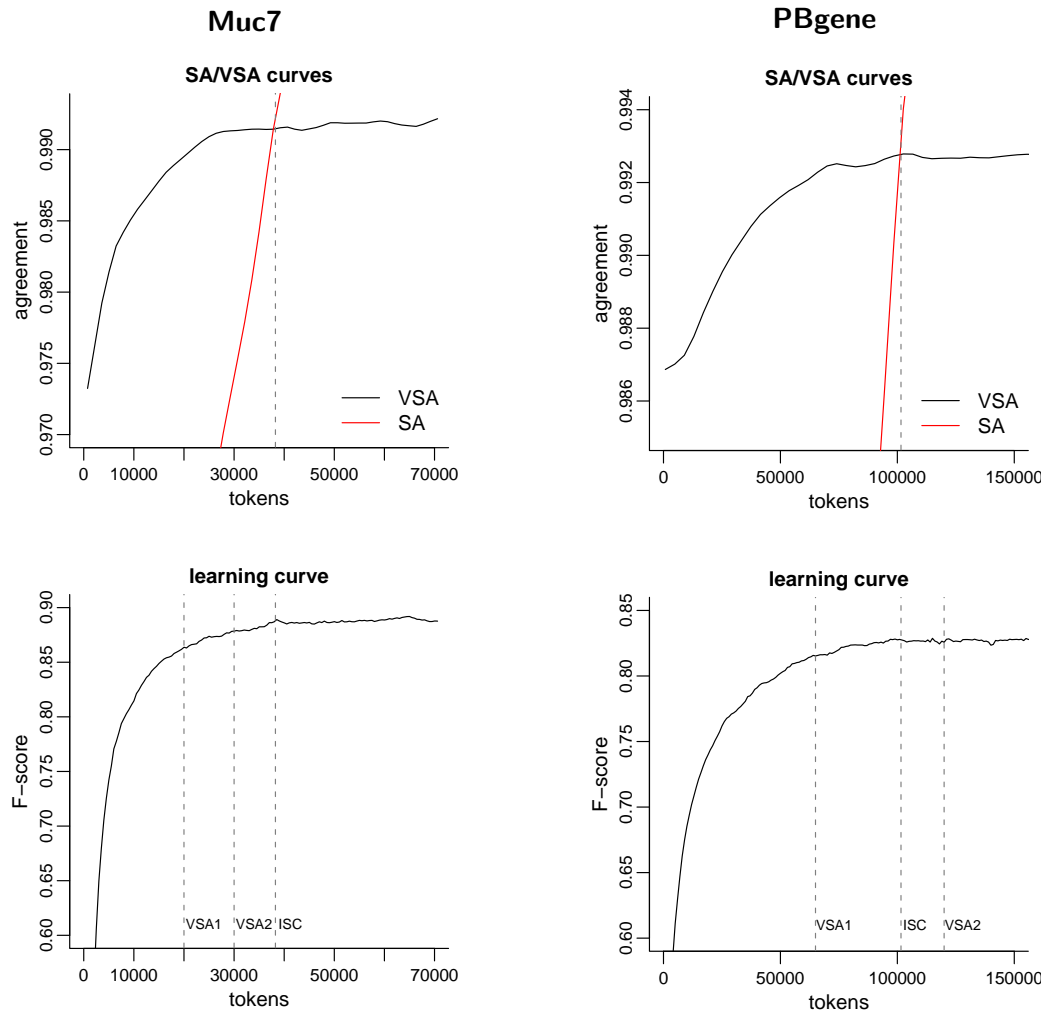


Figure 5.5: Stopping points (VSA1, VSA2, and ISC) shown on learning and agreement curves.

AL on Streamed Data The ISC determines the point when the most useful examples remaining in the pool of unlabeled data are less useful for training a classifier than the examples in the held-out validation set are on average. This means that the classifier would learn more from a sufficiently large sample taken from the validation set than it would if the AL process continued on the remaining unlabeled pool.

Besides the application as a conservative stopping criterion in the pool-based AL scenario, another practical scenario of the ISC is its application in a setting with dynamic pools, i.e., pool that change over time. As an example, assume that we are collecting data from a stream, for items from a news feed. Thus, the data is not available in the form of a closed set, but rather an open one which grows over time. To make the most of the human annotators in this scenario, we want them to operate on partitions of data instead of annotating individual news items as they are published. To do so, we wait until a given number of sentences have appeared on the stream, and then collect those sentences.

The problem is, how do we know when the AL-based annotation process for each such partition should be terminated? We clearly do not want the annotators to annotate all sentences, and we cannot have the annotators define trade-offs or set thresholds pertaining to the absolute classifier performance of each new partition of data available. By using the ISC, we are able to automatically issue a halting of the AL process and proceed to the next partition of data without losing too much in performance, and without having the annotators mark up too much of the available data. To this end, the ISC appears to be a reasonable decision help to find when more useful information can be gathered by switching to a new pool.

To carry out this experiment, we took a subsample of 10% (about 1,400 sentences) from a original AL pool of the CONLL corpus as a validation set.⁷ The rest of this pool was split into partitions of about 500 consecutive sentences. Committee-based AL with u_{VE}^s was now run taking the first partition as a pool to select from. At the point where the SA and VSA curve crossed, we continued AL selection from the next partition and so forth. Figure 5.6 shows the resulting learning curve. The intersection between the SA and VSA curves for each partition corresponds to the respective “steps” in the stair-like learning curve. Each intersection marks the point where we turned to the next partition.

The application of ISC in stream-based AL as an indicator for switching to new stream data is described in more detail in Olsson and Tomanek (2009).

⁷Note that the original CONLL test set was not used in this experiment, thus the F-score reported in Figure 5.6 cannot be compared to those reported before for the CONLL corpus.

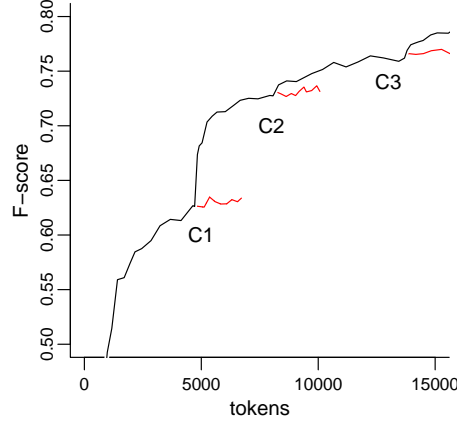


Figure 5.6: Learning curves for stream-based AL. C_i denotes the point at which AL is terminated for partition i and a new partition is employed instead. The red lines show how the learning curve on the old partition would have proceeded.

5.4 Summary

Due to our observation that the optimal stopping point is highly subjective and depends on application-specific preference structures between costs and benefits, we have proposed approximating the learning curve in an unsupervised manner, to which the subjective trade-offs can then be applied.

Our experiments primarily suggest that monitoring activities of AL processes should be done on a separate, held-out validation set that is not changed during the AL-based annotation set to avoid simulation artifacts. Based on such a validation set, the learning curve is approximated by calculating the $u_{VE}^{\bar{s}}$ utility scores on this set and plotting these scores over time.

While we found that the general progression of the learning curve can be approximated well by this approach, this does not include the actual slope or intercept. From such an approximation one can, however, tell whether AL is in the start-up, the transition, or the convergence stage. Absolute values of classifier performance can, however, not be read from it. Additionally, we proposed an explicit stopping criterion which is meant to be a conservative stopping point and does not require any user-defined thresholds or parameters.

Our comparative evaluation of stopping points derived from both manually applying subjective preference information and the ISC and SP stopping criteria empirically showed that the ISC is indeed a very conservative stopping point. This stopping

point is most often very close to the point where (almost) no further increase in performance can be reached. However, the ISC exhibits good savings of annotation effort compared to the position of the maximum performance, making ISC an appropriate stopping points in a preference-agnostic situation. Finally, this chapter discussed how the ISC can be applied in an AL setting with streamed data to find the point at which to switch from one data partition to the next. Each partition is assumed to be an accumulated portion of data from the stream.

Chapter 6

Semi-Supervised Active Learning

Approaches to AL, as discussed so far, considered unlabeled examples as atomic units for which, when selected, a complete annotation is required. Standard AL does not consider the internal structure of the selected examples. This may be a shortcoming especially when the AL selection granularity is coarse-grained, as in our NER scenario where complete sequences of words (here sentences) are selected.

Although a high overall utility may be attributed to a sequence as a whole, such a sequence may still exhibit subsequences that do not add up to the overall utility score. Regions of local model confidence may not require human labeling – the model can assign highly accurate labels itself. Figure 6.1 shows a sample sentence. The model of the current AL iteration is highly confident in its predictions on all tokens except “Shanghai”. Why should an annotator go through the complete sentence if there is only a single token on which the model requires human labeling support?

In sequence labeling scenarios of segmentation tasks, a common characteristic is that there are a few high-frequency, easy-to-learn classes combined with several low-frequency, hard-to-learn classes. This is, for example, the case in part-of-speech tagging, where determiners constitute such an easy, high-frequency-class. Similarly, this applies to the NER task, where larger stretches of a sentence or complete clauses do not contain any entity mention at all, or merely trivial instances of an entity class easily predictable by the current model.

This essentially leads to the assumption that the utility scores on subsequences are subject to a highly-skewed distribution and that such a characteristic of the data can be further exploited to reduce the human annotation effort per selected sequence.

Mystery	has	recently	surrounded	the	Shanghai	stockpile	...
0.01	0.001	0.0	0.00	0.02	0.495	0.003	...

Figure 6.1: Sample sentence for which a high overall utility score was obtained. The numbers below the words indicate the model’s confidence in its prediction.

This chapter presents an approach to semi-supervised AL for sequence labeling where selected sequences only need to be partially labeled by a human. The remaining, manually non-labeled parts are tackled in a semi-supervised learning manner.

After a short review of previous work on semi-supervised learning and previous approaches to combining AL and semi-supervised learning, we elaborate on the idea of semi-supervised AL for sequence learning and present a highly effective approach to it. A subsequent evaluation shows that extra annotation effort can be saved and explores the limitations of our approach.

Most of this work has been previously published in Tomanek and Hahn (2009b).

6.1 Related Work

Besides AL, semi-supervised learning is another approach to reducing the number of labeled training data. The core idea is to make use of unlabeled data which are assumed to be available at no cost. Semi-supervised learning is a lively field of research with a lot of study directions actively investigating in it (Chapelle et al., 2006). *Bootstrapping* is a widely known and general technique of iteratively increasing upon a given seed set of labeled data. In every step, additional labeled data is created based on the knowledge already available from previous steps and then added to the data set. Although bootstrapping may yield positive results under special conditions, it often fails. This is mostly due to the general problem that errors from automatic tagging propagate and damage the quality of the data set.

Self-training (Yarowsky, 1995) is an approach to semi-supervised learning based on bootstrapping. From a seed set of labeled examples a weak model is learned, which is subsequently incrementally refined. In each step, unlabeled examples on which the current model is highly confident are labeled with their predictions, added to the training set, and a new model is learned. Similar to self-training, co-training augments the training set with automatically labeled examples (Blum and Mitchell, 1998). Co-training is based on multiple learners having independent views on the data and mutually producing labeled examples for each other.

A combination of active and semi-supervised learning was first proposed by McCallum and Nigam (1998) for text classification and QbC-based AL. In each AL iteration, the committee members are at first trained on the labeled examples and then augmented by means of expectation maximization (EM) on unlabeled examples (Dempster et al., 1977). The idea is to avoid manual labeling of examples which may improve the model without having a label using EM. Similarly, co-testing (Muslea et al., 2002), another multi-view AL algorithm, selects examples for the multi-view,

semi-supervised Co-EM algorithm. In both works, semi-supervision is based on variants of the EM algorithm in combination with *all* unlabeled examples from the pool. Our approach to semi-supervised AL is different: we augment the *training data* using a self-tagging mechanism, while McCallum and Nigam (1998) and Muslea et al. (2002) used semi-supervision to augment the *models* based on the EM algorithm.

Tür et al. (2005) also presented a combination of AL and self-training. A general difference to our work is that in their approach examples were either fully labeled (when selected by AL) or completely automatically labeled in a self-training fashion (when subject to high confidence values). In our approach, sequence-based examples can be labeled both partially manually and automatically.

All semi-supervised AL approaches mentioned above have in common that they are based on a classification task. Our scenario is different in that it is about relational learning where the selection granularity is a sequence. This may give rise to local regions of higher or lower uncertainty so that partial manual labeling of selected sequences is reasonable. Along these lines, another work more closely related to ours is that of Kristjansson et al. (2004). In an IE application, the confidence per extracted field is calculated by a constrained variant of the Forward-Backward algorithm. Unreliable fields are highlighted so that the automatically annotated corpus can be manually *corrected*. In contrast, AL *selection* of examples together with partial manual *labeling* of the selected examples are the main foci of our approach.

6.2 Combination of Active Learning and Bootstrapping

Our approach to semi-supervised AL for sequence classification is based on bootstrapping as one form of semi-supervised learning. Bootstrapping approaches alone often fail to produce good results due to their inherent tendency to reduce the quality of augmented data by lots of tagging errors. This is especially the case in NLP tasks where large amounts of training material are required to achieve acceptable performance levels. Starting from a small seed, many iterations have to be run until a sufficient amount of training material is available. In the meantime, tagging errors cumulate and reinforce themselves.

Pierce and Cardie (2001) showed that the quality of the automatically labeled training data is crucial for co-training to perform well. In contrast, given too many tagging errors, one cannot learn a high-performing model. Also, the size of the seed set is an important parameter. When too small a seed set is chosen, data quality deteriorates quickly, when it is too large, no improvement over the initial model can be expected. To address the problem of data pollution by tagging errors, Pierce and Cardie (2001) proposed corrected co-training. In this mode, a human is put

into the co-training loop to review and, if necessary, to correct the machine-labeled examples. Although this effectively evades the negative side effects of deteriorated data quality, one may find the correction of labeled data to be as time-consuming as annotations from the scratch. The human annotator should not become biased by the proposed labels. Instead, she should independently examine an already labeled example so that correction eventually becomes annotation.

Our approach to semi-supervised AL cautiously incorporates bootstrapping by explicitly pointing human annotators to classification-critical regions of the selectively sampled examples. Such regions then require manual annotation, while regions of high confidence are automatically labeled and do not require any manual inspection. Under this directive, our approach to semi-supervised AL avoids deterioration of data quality while still selecting highly useful examples. Overall, we assume this procedure to achieve an extra reduction of human annotation effort compared to standard, fully-supervised AL, where selected sequences require full annotation.

6.2.1 Example Selection

In this chapter we assume an example to always be a sequence, such as a sentence for the NER scenario. For our approach to semi-supervised AL, the actual selection of examples is performed by standard AL, as described in Algorithm 2 on page 55, with the goal of finding examples with an overall high utility score as we expect to learn most from these examples.

Comparing this selection principle to the one underlying bootstrapping methods, such as self-training or co-training, it shows that bootstrapping would select sequences of *high confidence* to keep tagging errors as low as possible. This is essentially the opposite of AL-based selection. However, focusing on high-confidence examples, this approach will probably miss the really useful, unlabeled examples but still catch some tagging errors. In case of co-training, Pierce and Cardie (2001) require humans to review examples of limited learning utility.

Any of the utility functions presented in Section 4.2.3 may be applied to AL-based example selection here. However, we prefer utility functions based on the sequence-confidence, such as u_{LC}^s over those based on the aggregation of token-confidence because sequence-confidence based utility functions have an inherent tendency to select longer sequences as shown in Section 4.4.2.2. While longer sequences are expensive when the complete sequence needs to be labeled, they become rather inexpensive when the annotator is pointed to the few relevant subsequences. Thus, complete clauses on which the model is highly confident in its prediction can be

omitted. Short sequences of high utility are less likely to exhibit easy passages which do not need human judgement.

6.2.2 Identification of Critical Subsequences

After selection of a useful example $p = (\vec{x})$ with $\vec{x} = (x_1, \dots, x_n)$, the subsequences $\vec{x}' = (x_a, \dots, x_b)$, $1 \leq a \leq b \leq n$, of low local utility need to be identified. When no information is given on segmentation of a complete sequence into reasonable units, such as, for example, linguistically motivated phrases, we fall back onto subsequences of length 1 so that $\vec{x}' = (x_i)$ with $1 \leq i \leq n$. In application to NLP tasks, such a one-element subsequence is usually a single token. For the rest of this chapter, we assume such a token-level scenario. Extensions to longer subsequences are straightforward.

For a token x_i from a selected sequence \vec{x} we estimate the model's confidence $C_\theta(y_i^*)$ in label y_i^* . For a CRF, token-level confidence is given by the marginal probability (cf. Equation 2.31 on page 22) so that

$$C_\theta(y_i^*) = P_\theta(y_i = y_i^* | \vec{x}) \quad (6.1)$$

where y_i^* specifies the label at the respective position of the most likely label sequence \vec{y}^* obtained by the Viterbi algorithm (cf. Equation 2.29 on page 21). If $C_\theta(y_i^*)$ exceeds a certain confidence threshold t , y_i^* is assigned as the putatively correct label to x_i . Otherwise, manual annotation of this token is required.

In practise, sequences of consecutive tokens x_i with $C_\theta(y_i^*) \leq t$ should be presented to the annotator instead of individual tokens. For NLP tasks, one might consider linguistically motivated subsequences, such as noun phrases, which a human has to annotate when their confidence score falls below a threshold. Besides the one-element sequence, which is taken as our standard scenario here, we also evaluate our approach to semi-supervised learning on such linguistically motivated subsequences.

6.2.3 Parameters

Two parameters can be specified for our approach: firstly, the confidence threshold t which directly influences the portion of tokens to be manually labeled. Using lower thresholds, the self-tagging component has higher impact – presumably leading to higher numbers of tagging errors. Secondly, a delay rate d can be specified which channels the amount of manually labeled tokens obtained by standard, fully-supervised AL before semi-supervised AL is to start. Only with $d = 0$ will semi-supervised AL already affect the first AL iteration. Otherwise, several iterations of standard AL are run until a switch to semi-supervised AL occurs.

It is well known that the performance of bootstrapping approaches crucially depends on the size of the seed set, i.e., the number of labeled examples available to train the initial model. If class boundaries are poorly defined because too small a seed set is chosen, a bootstrapping system cannot learn anything reasonable due to high error rates. If, on the other hand, class boundaries are already too well defined due to an overly large seed set, nothing more can be learned. Together with low thresholds, we assume a delay rate of $d > 0$ to be crucial to obtain models of high performance.

6.3 Experiments and Results

Our approach to semi-supervised AL is evaluated for NER. By the nature of this task, the sentences are only sparsely populated with entity mentions and most of the tokens belong to the OUTSIDE class so that partial labeling is very beneficial.

The experiments compare our approach to semi-supervised AL (SESAL) to its fully-supervised counterpart (FUSAL) as well as random sampling where examples are completely labeled (RD). SESAL is first applied in a default configuration with a very strict confidence threshold of $t = 0.99$ and without any delay so that $d = 0$. In further experiments, these parameters are varied to study their impact on SESAL's performance. All experiments start from the default seed sets of 20 randomly selected examples. In each iteration, 50 new examples are selected by the u_{LC}^s utility function. The experiments were run on the MUC7 and the PBGENE corpus. For evaluation, the number of manually labeled tokens is taken as cost measure here.

6.3.1 Distribution of Confidence Scores

The basic assumption about non-uniform, highly skewed distribution of the token confidence scores means that only a small proportion of tokens within the selected sentences constitute really hard decision problems, while the majority of tokens are easily accounted for by the current model. To test this stipulation, we investigate the distribution of the model's confidence values $C_\theta(y_i^*)$ over all tokens of the sentences selected within one iteration of FUSAL. Figure 6.2, as an example, depicts the histogram for an early AL iteration round on the MUC7 corpus. The vast majority of tokens has a confidence score close to 1 and the median lies at 0.9966. Histograms of subsequent AL iterations are very similar, with an even higher median. This is so because the model becomes continuously more confident when trained on additional data and fewer hard cases remain in the shrinking pool.

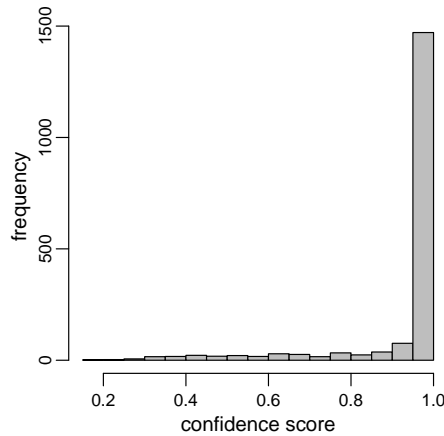


Figure 6.2: Distribution of token-level confidence scores in an early AL iteration.

	MUC7 F = 0.88	PBGENE F = 0.83
$\Delta\text{CFP}_{F^*}(\text{SESAL}, \text{FUSAL})$	69.5 %	67.2 %
$\Delta\text{CFP}_{F^*}(\text{SESAL}, \text{RD})$	82.5 %	85.9 %

Table 6.1: Percentage reduction of annotation effort in terms of *manually labeled tokens* to yield that target performance of F=0.88 on MUC7 and F=0.83 on the PBGENE corpus.

6.3.2 Fully-Supervised vs. Semi-Supervised AL

Figure 6.3 compares the performance of FUSAL and SESAL on the two corpora. SESAL is run with a delay rate of $d = 0$ and a very high confidence threshold of $t = 0.99$ so that only those tokens on which the current model is almost certain are automatically labeled. This figure clearly shows that SESAL is much more efficient than its fully-supervised counterpart. Table 6.1 depicts how much annotation effort is reduced by SESAL to reach the maximal F-score on both corpora.¹ While FUSAL itself saves about 50 % compared to RD, SESAL saves up to 69.5 % compared to FUSAL which constitutes an overall saving over RD of over 82 %.

Figure 6.3 reveals that the learning curves of SESAL stop early: after 12,800 tokens on the MUC7 tokens corpus and after 27,600 tokens on the PBGENE corpus. This early termination is because at that point the whole corpus has been exhaustively labeled – either manually, or automatically. So, using SESAL, we can come up with

¹For an overview of the maximal, passive learning F-scores on all corpora see Table 4.3.

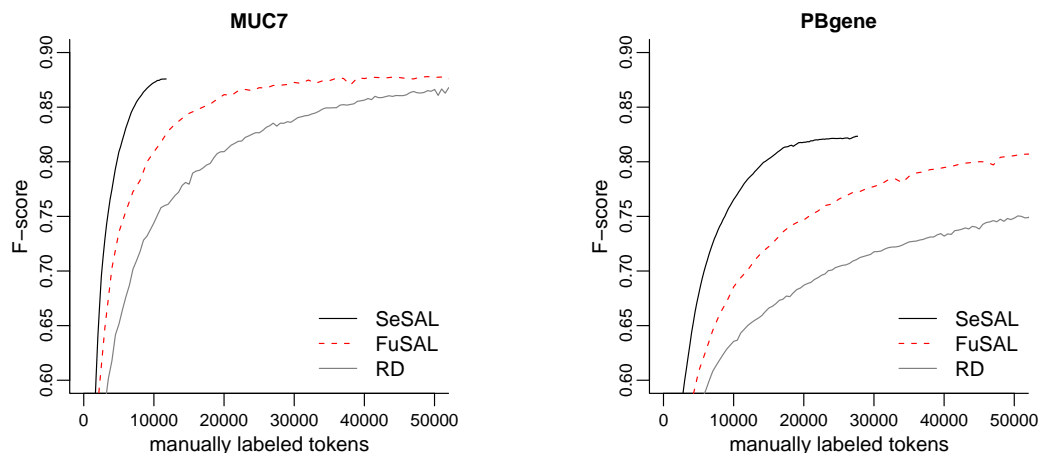


Figure 6.3: Learning curves for semi-supervised AL (SeSAL), fully-supervised AL (FuSAL), and random selection (RD).

a fully-labeled corpus of which only a small fraction actually needs to be manually labeled (about 18 % on MUC7 and 13 % on PBGENE).

6.3.3 Annotated Noun Phrases

The savings by SESAL mentioned so far were calculated relative to the number of *tokens* that have to be manually labeled. However, imagine the following situation. Assume that, using SESAL, every second token in a sequence would have to be labeled. Although this comes to a formal saving of 50 %, the actual annotation effort in terms of the *time* needed would hardly decline. It appears that only when SESAL splits a sentence into larger, chunk-like subsequences can annotation time really be saved.

To demonstrate that SESAL comes close to this, we counted the number of noun phrases containing one or more tokens to be manually labeled. Table 6.2 shows the results for this experiment. On both corpora, SESAL saves about 45 % in terms of

	MUC7	PBGENE
	F = 0.87	F = 0.81
$\Delta CFP_{F^*}(\text{SeSAL}, \text{FuSAL})$	44.5 %	45.8 %

Table 6.2: Percentage reduction of annotation effort in terms of *annotated noun phrases* to yield that target performance of F=0.87 on MUC7 and F=0.81 on the PBGENE corpus.

manually labeled	automatically labeled	AR	number of errors	ACC
1,000	253	79.82	6	0.995
5,000	6,207	44.61	82	0.993
10,000	25,506	28.16	174	0.995

Table 6.3: Characteristics of SESAL on the MUC7 corpus for a specific amount of manually annotated tokens. Annotation rate (AR) is the proportion of manually labeled tokens in the total amount of labeled. ACC refers to the accuracy of the labels in the corpus.

manually labeled	number of errors	error types (%)		
		E2O	O2E	E2E
10,000	75	100	–	–
70,000	259	96	1.3	2.7

Table 6.4: Distribution of self-tagging errors in percent on the MUC7 corpus. Error types: OUTSIDE class assigned though an entity class is correct (E2O), entity class assigned but OUTSIDE is correct (O2E), wrong entity class assigned (E2E).

the number of noun phrases touched, i.e., containing tokens with confidence below the confidence threshold t .

6.3.4 Selection Characteristics of Sesal

Table 6.3 shows of characteristics the corpus created by SESAL. In very early AL rounds, a large proportion of tokens has to be manually labeled (70-80 %). This number decreases as the classifier improves and the pool contains fewer informative sentences. The number of tagging errors is quite low, resulting in a high level of accuracy of the created corpus of consistently above 99 %.

The majority of the automatically labeled tokens (97-98 %) belong to the OUTSIDE class. This coincides with the assumption that SESAL works especially well for labeling tasks where some classes occur predominantly and can, in most cases, be discriminated easily from the other classes, as is the case in the NER scenario. An analysis of the errors induced by the self-tagging component reveals that most of the errors (90-100 %) are due to missed entity classes – while the correct class label for a token is one of the entity classes, the OUTSIDE class was assigned. This effect is more severe in early than in later AL iterations. Table 6.4 provides exact numbers.

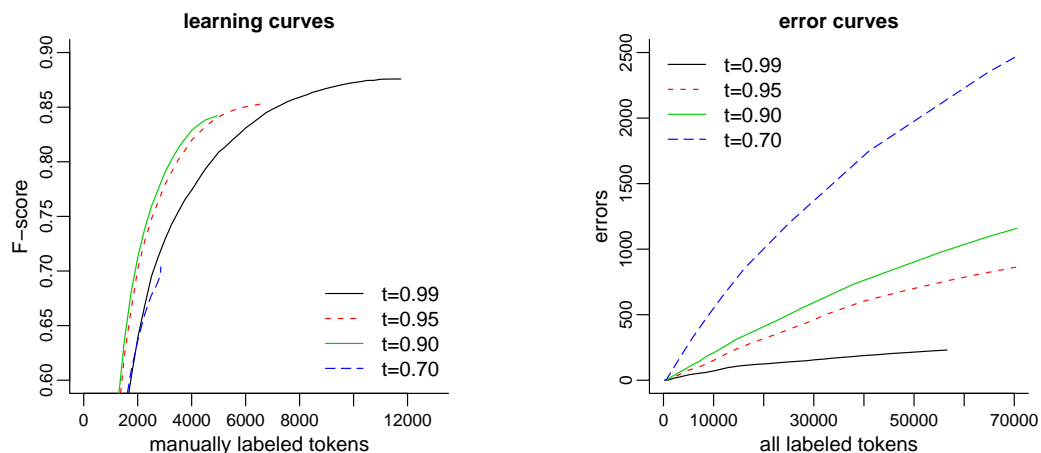


Figure 6.4: Learning and error curves for SESAL with different confidence thresholds on the MUC7 corpus.

threshold	F-score	Recall	Precision	ACC
0.99	0.88	0.86	0.90	0.996
0.95	0.85	0.82	0.89	0.998
0.90	0.84	0.81	0.88	0.984
0.70	0.70	0.62	0.81	0.965

Table 6.5: Maximal model performance when the complete MUC7 corpus has been labeled with SESAL and different confidence thresholds.

6.3.5 Impact of the Confidence Threshold

As another experiment, SESAL was run with different confidence thresholds t (0.99, 0.95, 0.90, and 0.70) and the results were analyzed with respect to tagging errors and the model performance. The motivating assumption for this experiment was that by using lower thresholds fewer tokens actually need to be manually annotated. Applying such less restrictive thresholds, SESAL becomes more similar to self-training resulting in higher numbers of tagging errors.

Figure 6.4 shows the learning and error curves of SESAL for different thresholds. On MUC7, the maximal F-score of 87.7% is only reached by the highest and most restrictive threshold of $t = 0.99$. With all other thresholds, SESAL stops at much lower F-scores and produces labeled training data of lower accuracy (the same holds for PBGENE). Table 6.5 gives exact numbers.

Interestingly, the poor model performance of SESAL with less restrictive thresholds

is mainly due to falling recall values. This is because tagging errors are mainly those where the OUTSIDE class is assigned, although an entity class was correct (cf. Table 6.4). In consequence, in the corpus resulting from SESAL, several entity mentions are not annotated as such so that the model trained on this corpus is ignorant about such entity mentions as well.

6.3.6 Impact of the Delay Rate

We also measured the impact of delay rates on SESAL’s efficiency considering three delay rates (1,000, 5,000, and 10,000 tokens) in combination with three confidence thresholds (0.99, 0.9, and 0.7). Figure 6.5 depicts the respective learning curves. On both corpora, for SESAL with $t = 0.99$, the delay has no particularly beneficial effect. However, in combination with lower thresholds, the delay rates show positive effects as SESAL yields F-scores closer to the respective maximal F-score.

6.3.7 Discussion

Our experiments in the context of the NER scenario provide empirical evidence that our proposed approach to semi-supervised AL for sequence labeling does indeed considerably reduce the amount of tokens to be *manually* annotated — in terms of numbers, almost 70% compared to its fully-supervised counterpart, and over 80% compared to a totally passive learning scheme based on random selection. Thus, semi-supervised AL successfully joins the standard, fully-supervised AL schema with a bootstrapping mode, namely self-training, to combine the strengths of both approaches.

Moreover, a model trained on a corpus created with semi-supervised AL can have the same performance compared to a model trained on the fully manually-labeled counterpart. Thus, semi-supervised AL effectively overcomes the problem of deteriorated data quality which is a major problem of many bootstrapping approaches. For semi-supervised AL to work well, a high and, as a result, restrictive threshold has been shown to be crucial. Otherwise, large numbers of tagging errors lead to a poorer overall model performance. The delay rate is important for scenarios with low confidence thresholds because early tagging errors can be avoided which otherwise reinforce each other. Finding the right balance between the delay rate and low thresholds requires experimental calibration. However, for the most restrictive threshold ($t = 0.99$), such a delay is superfluous so that it can be set to $d = 0$, circumventing this calibration step.

In summary, the self-tagging component of SESAL gets more influential when the confidence threshold and the delay rate are set to lower values. At the same time though, under these conditions negative side-effects such as deteriorated data quality and, in consequence, inferior models emerge. Our experiments indicate that as long as self-training is cautiously applied (as is done for SESAL with restrictive parameters), it can definitely outperform an entirely supervised approach. With the confidence threshold, the SESAL approach has a means to execute control over the influence of self-training.

6.4 Summary and Conclusions

We have presented an approach to semi-supervised learning which is based on the idea that only regions of high uncertainty within a selected sequence are manually selected – the rest is automatically tagged. According to our evaluation, this procedure constitutes an extra reduction of annotation effort over standard AL where the complete sequence is manually labeled: above 67 % in terms of tokens manually labeled and about 45 % in terms of noun phrases manually labeled.

From an annotation point of view, our approach to semi-supervised AL efficiently guides the annotator to regions within the selected sentence which are very useful for the learning task. In our experiments on the NER scenario, those regions were mentions of entity names or linguistic units which had a surface appearance similar to entity mentions but could not yet be correctly distinguished by the model. We hypothesize that human annotators work much more efficiently when pointed to the regions of immediate interest instead of skimming in a self-paced way through larger passages of probably semantically irrelevant but syntactically complex utterances – a tiring and error-prone task.

At this point, an open question is whether the reported savings translate to the actual annotation time needed. A detailed evaluation of semi-supervised AL in terms of true annotation time is given in Chapter 11 on cost-sensitive AL.

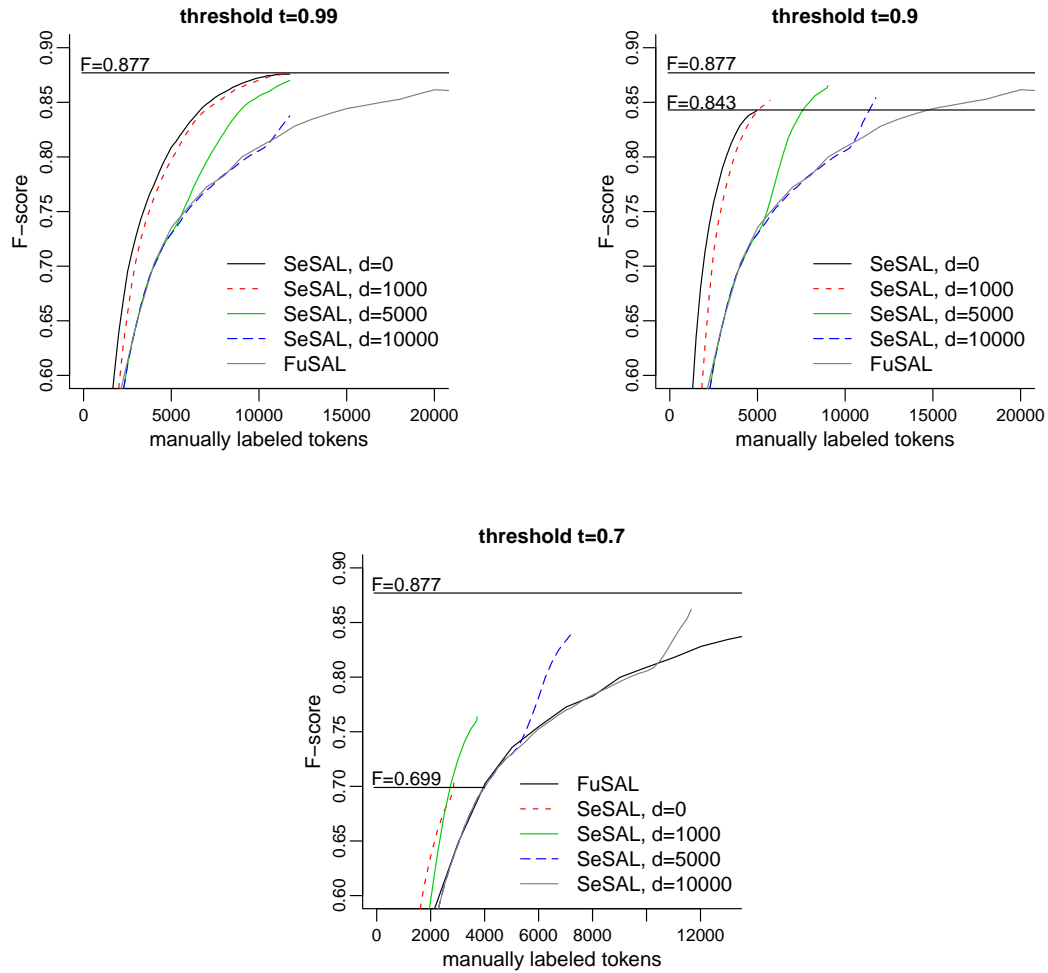


Figure 6.5: SESAL with different delay rates and thresholds on MUC7. Horizontal lines mark the supervised F-score (upper line) and the maximal F-score achieved by SESAL with the respective threshold and $d = 0$ (lower line).

Chapter 7

Sample Efficiency in Scenarios of Disparate Selector-Consumer Pairings

Previous chapters have shown that AL does indeed have a high potential for reducing the burden of annotation. While the reported savings in annotation effort certainly encourage the use of AL as a novel, resource-aware annotation strategy, skeptics warn about the sampling bias induced by AL.

By definition of AL, the data distribution of the sample obtained by AL is intended to diverge from the “true” underlying data distribution – as a selective sampling method, AL is based on a bias towards useful examples instead of less useful ones. While a sampling bias towards useful examples is desirable, one must keep in mind that the utility is defined with respect to the selector applied during AL selection and that the resulting data is additionally biased towards a particular learner.

Experiments performed so far for this thesis have in common that selector and consumer are based on the same model. This correspondence of selector and consumer is tacitly assumed in almost all studies of AL. However, there are scenarios where selector and consumer diverge.

- One might apply a less complex model as selector while the final consumer remains the high-accuracy, more complex one. This is motivated by the fact that AL selection time depends to a great extent on the computational complexity of training the selector in each AL iteration. Accelerated AL selection, as demanded by Requirement 2, can be yielded by the utilization of less complex to train models during selection.
- At the time of data acquisition the consumer to be finally used is often unknown. This is the case when training data for new learning problems has to be created. In consequence, the learning algorithms used during selection will most likely be different from the final consumer. The difference may be in terms of the model chosen, but also with respect to the particular feature set employed.

In the first scenario, selector and consumer deliberately diverge to reduce the *computational complexity of sampling*. The second scenario is characterized by incomplete knowledge about the final consumer so *sample flexibility* needs to be increased. Both scenarios describe a setting where a sample is biased towards a particular model.

With respect to Requirement 5, this chapter investigates how sampling efficiency of AL is affected by scenarios where selectors and consumers are based on different models classes. We say that an AL sample is *reusable* by a particular consumer, if the consumer yields a higher classification accuracy on this sample than it would on a random sample.

The rest of this chapter is structured as follows: firstly, we review the general problem of learning from biased samples, then we define sample reuse and sample reusability in the context of AL, and study previous work on this issue. Section 7.3 empirically investigates how sampling efficiency of AL is affected by disparate selector-consumer pairings. This section also aims at identifying factors that positively or negatively affect AL sample reusability. Section 7.4 describes an approach to reduce negative effects of sample reuse and in this way to increase AL sample reusability. Finally, the last section summarizes our findings and identifies future directions of research.

7.1 Learning Under Sample Selection Bias

ML algorithms operate under the assumption that both training and test data obey to the same distribution. In practise, however, training data is often governed by a distribution dissimilar to that of the data to which the learned model will finally be applied. When training and test distributions differ this is generally referred to as *sample selection bias*.¹ Learning under sample selection bias is problematic because the model parameters optimized on the training data may be suboptimal when the model is applied to test data with a dissimilar distribution.

Zadrozny (2004) introduced a categorization of different types of sample selection bias including class bias (selection process dependent on class labels y , only), feature bias (selection process dependent on features f_j , only), and finally complete bias (selection process dependent on y and x). Feature bias, also known as covariate shift, is one of the most common types of biases.² It is defined as a scenario where the marginal distributions of training and test data diverge, $P_{\mathcal{L}}(\vec{x}) \neq P_{\mathcal{T}}(\vec{x})$, but

¹ Sample selection bias has originally received lots of attention in econometrics where data is often collected through surveys and thus is naturally subject to the self-selection bias. Self-selection biased occurs when people can decide whether to take part in a survey or not. In this context, Heckman (1979) proposed an approach to correct the bias.

²It should be noted that “covariate shift” is usually used in the context of regression learning.

structural relations remain the same so that the conditional distribution of target labels is unchanged and $P_{\mathcal{L}}(y|\vec{x}) = P_{\mathcal{T}}(y|\vec{x})$ (Shimodaira, 2000).

Computational learning under sample selection bias has recently received great attention.³ Most approaches to cope with sample selection bias are based on the idea of re-weighting training examples according to their importance in the test data (Cortes et al., 2008). The weighting factor is often estimated as a density ratio factor $w(x) = \frac{P_{\mathcal{T}}(\vec{x})}{P_{\mathcal{L}}(\vec{x})}$ and the particular approaches to correct bias basically differ in the way $w(x)$ is estimated (Shimodaira, 2000; Bickel et al., 2007; Tsuboi et al., 2009).

Though most work on sample selection bias has focused on approaches to correct the bias, another crucial question is how sensitive learners are to the bias. Zadrozny (2004) and Fan et al. (2005) studied well-known models – including C4.5, SVM, NB, and MaxEnt – according to their sensitivity to sample selection bias. An inductive learner is called *local* if it is invariant to bias, and *global* otherwise. While Zadrozny (2004) categorized the considered learning algorithms independent of the actual classification problem, Fan et al. (2005) relaxed this strict categorization and showed analytically and empirically that the considered learners can be both local and global depending on the combination of the data set, modeling assumptions of the learner and their appropriateness to model the particular data set.

An important issue in this context is whether a learning algorithm can find a model θ^* in the model space Θ so that the estimated probability $P_{\theta^*}(y|x)$ is equivalent to the true model $P(y|x)$. A learner is called *consistent* if such a θ^* exists. It has been argued that covariate shift does not affect learning asymptotically, given a consistent learner (Shimodaira, 2000; Fan et al., 2005). In practise, and for complex problems such as most NLP tasks, it is, however, very unlikely that there is a consistent learner so that sample selection bias does indeed affect learning.

Sample selection bias is an ubiquitous problem in many applications (Tsuboi et al., 2009). Most NLP problems are naturally subject to sample selection bias because lexical information is amongst the most relevant features and the lexical distribution of the training data generally differs from that of the test data (Son et al., 2009).

7.2 Sample Reuse as a Case of Sample Selection Bias

Sample selection bias is also a natural companion of selective sampling strategies such as AL. In the previous section we implicitly assumed that sample selection bias does affect learning in a negative way. In contrast, the core idea of AL is the

³This is for example evidenced by the NIPS 2006 Workshop on “Learning when test and training inputs have different distributions” (Airoldi et al., 2006).

induction of a *favorable* bias so that a particular learner can derive a good model with fewer training data.

7.2.1 Sample Reuse and Reusability

In the context of AL, the question of interest is whether the bias induced by AL with a particular selector positively or negatively affects learning given another learner for the final model.

Definition 12 (*Sample Reuse*) *AL sample reuse describes a scenario where a sample S obtained by AL using learner T_1 during selection is exploited to induce a particular model type with learner T_2 with $T_2 \neq T_1$.*

In the standard AL scenario we assume selector and consumer to be induced by the same learner, i.e., be based on the same model type. In previous experiments in this thesis, for example, a CRF-based model was used both as selector and consumer. We call this scenario AL *self-selection*. In contrast, AL *foreign-selection* specifies a scenario of sample reuse.

In the AL setting, the appropriateness of a sample is quantified by the classification accuracy of the consumer induced from this sample. As sampling efficiency of AL is typically compared to random sampling, the performance obtained on a random sample constitutes a lower bound for sampling efficiency which selective sampling needs to exceed to be considered efficient. However, one would expect that a sample that is optimal for one learner does not necessarily outperform a random sample when used by another learner.

Definition 13 (*Sample Reusability*) *Given a random sample S_{RD} , a sample S_{T_1} obtained with AL and a selector based on learner T_1 , and a learner T_2 with $T_2 \neq T_1$. We say that S_{T_1} is reusable by learner T_2 if a model θ' learned by T_2 from this sample, i.e., $T_2(S_{T_1})$, exhibits a better performance on a held-out test set \mathcal{T} than a model θ'' induced by $T_2(S_{RD})$, i.e., $\text{perf}(\theta', \mathcal{T}) > \text{perf}(\theta'', \mathcal{T})$.*

While sample selection bias induced by AL self-selection is normally beneficial for learning, the bias induced by AL foreign-selection is expected to constitute a bias inferior to the self-selection bias. Whether the AL foreign-selection bias affects learning so negatively that performance drops below the random selection baseline is one of the central research questions in this chapter.

Figure 7.1 illustrates sample reuse and reusability. The learning curves refer to samples obtained by AL with selectors based on different learners T_i or random

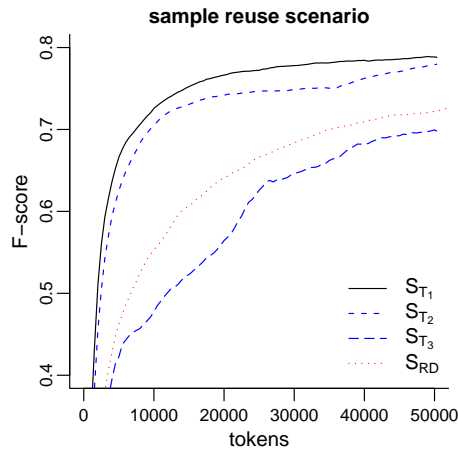


Figure 7.1: Sample reuse scenario where a learner T_1 is used as a consumer to obtain the learning curves. Selectors used during AL are based on learners T_1 , T_2 , and T_3 . S_{T_1} constitutes an AL self-selection scenario, S_{T_2} and S_{T_3} are AL foreign-selection scenarios. Sample S_{T_2} is reusable by T_1 , S_{T_3} is not (below random sampling).

sampling (RD). All samples are evaluated with a consumer based on learner T_1 . The dotted line corresponds to the performance of a model induced by $T_1(S_{RD})$. The solid curve shows an AL self-selection scenario where both selector and consumer are based on learner T_1 . The two dashed lines correspond to AL foreign-selection scenarios with selectors based on learners T_2 and T_3 , respectively. While S_{T_2} is reusable by T_1 (above the random selection baseline), the bias contained in S_{T_3} is too adversarial for T_1 , so that S_{T_3} is not reusable by T_1 (below the random baseline).

Sample reuse in the context of AL is of high practical relevance. Given sample reusability, a fast learner may be used during selection to produce training material well-suited to a more complex learner. Reduction of the computational complexity of sampling is especially attractive when AL is applied to NLP problems. For instance, CRFs generally perform very well on the NER task but exhibit an excessively high training times.⁴ On the other hand, reusability enables sample flexibility for scenarios where the best model is not known prior to the annotation process.

Sample reusability, on the other hand, is not necessarily given. We have learned in the previous section that sample selection bias in combination with inconsistent models may be adversarial. Transferred to the AL sample reuse scenario, *adversarial*

⁴On a 3.00Ghz Intel dual-core processor, 10 fold training of a CRF on the PBGENE corpus took about 66 minutes. In comparison, the MaxEnt learner took 15 and NB only 2 minutes. (Note that these times include generation of feature from plain training data.)

sample selection bias is defined as a scenario where a sample obtained by AL is less useful for classifier learning than a randomly drawn sample of the same size.

In this chapter, reusability for several learners and data sets is empirically studied. What is most important is the sensitivity of learner towards AL foreign-selection sample selection bias and the resulting presence or absence of reusability.

7.2.2 Previous Work

There has been only little work done on AL sample reuse and the issue of reusability. A scenario of sample reuse motivated by reduction of computational complexity of sampling was first described by Lewis and Catlett (1994). For text classification, they applied a decision tree to learn the consumer. During AL selection a logistic regression model is used. Lewis and Catlett reported positive findings about sample reusability. Additionally, they propose a method to correct the bias obtained by AL foreign-selection: A specific loss ratio is introduced to counterbalance bias towards the low frequency class which would otherwise harm a decision tree.

In the context of AL for SVMs, Vlachos (2004) reports on experiments of sample reuse with different SVM kernels (linear and RBF). Both kernels are used once each as the selector and the consumer with the result that better reusability is achieved when the stronger RBF kernel is used during selection. Vlachos' experiments might suggest that during AL selection the best performing learner should be employed in order to keep reusability high. Our experiments (see below) disagree with this observation as a general pattern.

For the NLP task of statistical parsing, controversial findings on reusability have been published. Hwa (2001) reported positive results and showed that a sample obtained by AL with a model of the Collins parser as selector, is reusable for a consumer based on another parser.⁵ In contrast, Baldridge and Osborne (2004) showed scenarios where the AL foreign-selection bias impairs reusability. They experimented with different parser models based on exchanged feature sets and models. In conclusion, they argued that AL should be used as a method to create labeled data only when the selector and the consumer are likely not to be substantially different as otherwise reusability of the created data can not be guaranteed.

Baldridge and Osborne (2004) hypothesized that the utility of the data selected by one learner for another learner depends on the degree of relatedness of the learned models under consideration. To empirically test this hypothesis they compare the

⁵The Collins Parser is a well-known statistical natural language parser (Collins, 2003). The second parser is based on Probabilistic Lexicalized Tree-Insertion Grammars (Hwa, 2001).

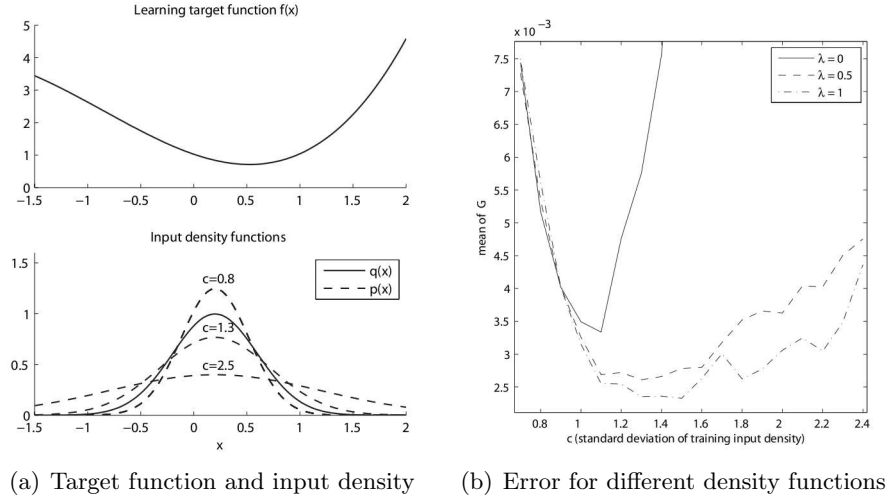


Figure 7.2: Effect on mean error of different model parameters λ in combination with different input densities. Plots taken from (Rubens and Sugiyama, 2006).

utility ranks assigned to the unlabeled examples by different models used as selectors with Spearman's rank correlation coefficient.⁶ Similar rankings lead to high correlation scores and are interpreted as high relatedness of the models. In their experiments, they found that high reusability is obtained when the selector and the consumer models are highly related according to this rank correlation score.

In line with the findings of Baldridge and Osborne (2004), one would intuitively expect that examples that constitute an optimal training set for one learner may be poor for another learner. Rubens and Sugiyama (2006) demonstrated this assumption in a toy example on regression learning. The one-dimensional training data is normally distributed ($\mu = c$, $\sigma = 0.4c$) as shown in Figure 7.2(a). Rubens and Sugiyama applied a special form of a least squares approach for regression learning in which a model parameter λ needs to be specified. Figure 7.2(b) nicely illustrates that the chosen model parameter λ crucially depends on the training input density. While for $c \leq 0.9$ models with different λ parameters exhibit approximately the same generalization error, for $c = 1.4$ a model with $\lambda = 0$ leads to a much higher error than a model with $\lambda = 1$. This emphasizes that training set density resulting from a selective sampling process has different effects on different models (obtained by different values of λ in this example).

These few papers available provide evidence for scenarios of presence and absence of AL sample reusability in the context of different learning problems. To date, there is no comprehensive study investigating the true nature of reusability, requirements

⁶Appendix D.1 shortly discusses how Spearman's rank correlation coefficient is calculated.

for the presence of reusability, or prohibitive factors. While this chapter does not intend to fill this gap completely, it at least provides a starting point for further investigations and places reusability in the context of learning under sample bias.

7.3 Empirical Investigation of Sample Reusability

Our investigation on AL sample reusability is driven by the following hypotheses:

H1: Limited Sample Reusability Given the work of Rubens and Sugiyama (2006), we have good reason to assume that sample selection bias through AL foreign-selection can be very adversarial. In combination with the work of Baldrige and Osborne (2004), which shows that AL foreign-selection can lead to non-reusable samples and the assertion that, in practise, learners are hardly consistent for a specific problem (Tsuboi et al., 2009), it leads us to assume that sample reusability is but a rare lucky chance and cannot be assumed in the general AL setting.

H2: Self-selection as Upper Bound AL self-selection constitutes a relevant but presumably positive sample selection bias. We assume AL self-selection to constitute the upper bound of sampling efficiency for all AL foreign-selection scenarios and AL foreign-selection to perform at its best as well as self-selection.

H3: Selector-Consumer Pairings exhibit General Reusability Characteristics Any combination of learners when used as selector-consumer pair in an AL foreign-selection setting might exhibit a particular reusability characteristic. We assume such reusability characteristics to be general for the specific selector-consumer pair so that they hold for all data sets and learning problems.

One such characteristic could be that AL with learner T_1 leads to a considerably different AL sample bias than learner T_2 , so that in consequence $\text{perf}(\theta', T) \ll \text{perf}(\theta'', T)$, where θ' is a model induced by $T_2(S_{T_1})$ and θ'' is induced by $T_2(S_{T_2})$. The other way round, two learners T_1 and T_2 might lead to the same sample bias so that they can be used interchangeably as selector or consumer for each other.

In the following section, a large-scale AL sample reuse experiment is performed to generate the empirical foundation to analyse the hypotheses. After testing hypotheses H1-H3, we turn to four follow-up hypotheses.

data set	examples	features	classes	missing values	feature types
CAR	1,728	6	4	no	categorical
MUSHROOM	8,124	22	2	yes	categorical
NURSERY	12,960	8	5	no	categorical
SEGMENT	2,310	19	7	no	real
SICK	3,772	30	2	yes	mixed

Table 7.1: Overview of selected data sets from the UCI repository.

7.3.1 Large-Scale Experiment on Sample Reuse

For the experiments on sample reuse we applied several models used in all combinations of selector-consumer pairs. To not constrain the experiments to a single learning problem, we performed experiments both on the specific task of NER and on several well-known, non-linguistic classification problems taken from the UCI Machine Learning Repository (Asuncion and Newman, 2007).

7.3.1.1 Experimental Settings

Data Sets For the NER scenario, sample reuse experiments were performed on the MUC7 and the PBGENE corpus so that our experiments cover one scenario with many and one with few entity classes (7 classes on MUC7 vs. 3 classes on PBGENE) and two different domains (newspaper vs. biomedical).⁷ From the UCI repository, which currently hosts about 120 data sets on classification problems, we picked five sets according to the following criteria: firstly, for AL experiments the data sets should have a reasonably large number of examples, so only data sets with more than 1,000 examples were considered; secondly, we wanted to have data sets with different characteristics according to the learning problem, the number of features, and the target classes. Table 7.1 gives an overview of the UCI data sets selected.

Learning Algorithms and Utility Functions Besides the linear-chain CRF, the following additional learners found application in the NER scenario: From the MALLET ML toolkit (McCallum, 2002), the NB and the MaxEnt learners were applied with exactly the same feature sets as the CRF. The linear kernel SVM from the LIBLINEAR implementation (Fan et al., 2008) was also applied with the same feature set. Finally, Lingpipe’s implementation of a HMM (Alias-i, 2008) with its own feature

⁷Preliminary experiments were also run on other NER corpora not reported on here with generally comparable results.

set was applied. The token-level confidence margin utility function $u_{\text{MA}}^{\bar{s}}$ (cf. Section 4.2.3) was used as the utility function for the CRF, the NB, and the MaxEnt learners.

For the SVM, the token-level margin-based utility function was calculated differently. LIBLINEAR follows a one-versus-rest approach for multi-class classification (cf. Section 2.2.2). For maximum margin classification, larger decision values $d(\vec{x})$ indicate that the example x is farther away from the hyperplane. Following Schölkopf and Smola (2002), larger distances can be interpreted as higher confidence of the classifier in its classification. For the multi-class SVM scenario, we applied the following margin-based utility function for US-based AL:

$$\begin{aligned} u_{\text{SVM}}(p, \vec{w}, b) &= -(d_{y^*}(\vec{x}) - d_{y^{**}}(\vec{x})) \\ y^* &= \operatorname{argmax}_{y' \in \mathcal{Y}} d_{y'}(\vec{x}) \\ y^{**} &= \operatorname{argmax}_{\substack{y'' \in \mathcal{Y} \\ y' \neq y''}} d_{y''}(\vec{x}) \end{aligned} \tag{7.1}$$

A NB-, a MaxEnt-, a Decision Tree-, and a linear kernel SVM-based learner are applied on the UCI data sets. We resort to the WEKA implementation (Witten and Frank, 2005) of all these classifiers and apply them in their default parameter settings.⁸ WEKA’s implementation of the C4.5 algorithm, called J48, is used for the Decision Tree learner. It is well known that decision trees are subject to the tree instability problem which means that small changes in the training set may cause dramatically large changes in the learned tree classifier. To circumvent this problem, a J48-based consumer is trained using bagging (Breiman, 1996).⁹ For all experiments on the UCI data set, features as specified in the data sets were used.

As for the AL experiments on the UCI data, the general AL framework as presented in Chapter 3 is used. The selection granularity here is a single example, in each iteration exactly the example with the highest utility score is selected.¹⁰ For the NB and the MaxEnt learners, the u_{MA} utility function is applied; AL with the SVM learner applies u_{SVM} defined in Equation 7.1. Due to the instability of decision trees, US-based AL cannot be applied efficiently for J48 (Dwyer and Holte, 2007). Instead, we employ QbC-based AL with the Vote Entropy utility function for J48.

AL experiments in the NER scenario are stopped after 50,000 tokens. On the UCI data sets, convergence occurs very early so that we stopped AL after 150 examples. The results reported in the following are averages over 20 independent runs.

⁸See Appendix D.2 for an overview of the parameter settings used.

⁹WEKA’s implementation of Breiman’s bagging approach was used with default parameters.

¹⁰Selection of the best example is feasible because the UCI learning problems are computationally less complex than experiments on the NER task due to the much lower number of features.

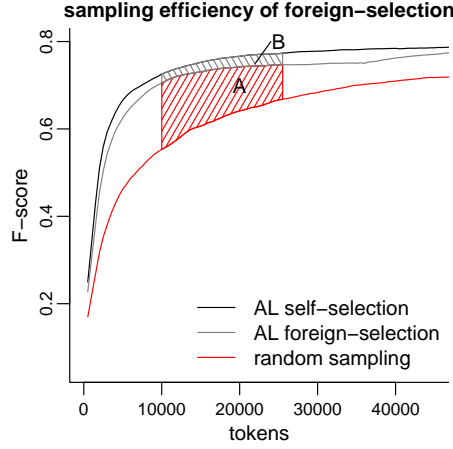


Figure 7.3: Quantification of sample reusability through the REU score which is here calculated by $\frac{A}{A+B} - 1$. In this example, $\text{REU} = -0.17$ indicates good reusability.

Quantification of Sample Reusability Previously, we considered reusability as a binary measure – reusability is either in evidence for a particular consumer or not. To quantify sample reusability on a continuous scale we introduce a novel measure. Similarly to the RAI measure (cf. Equation 4.14 on page 61) this measure is also defined on an interval $[a, b]$. Given a baseline sampling scenario S_{base} , typically random selection, the learning curve of AL self-selection S_{self} , and the learning curve of AL foreign-selection S_{frgn} , the REU score is defined as

$$\text{REU}(S_{\text{frgn}}, S_{\text{self}}, S_{\text{base}}, a, b) = \frac{\text{AUC}(S_{\text{frgn}}, a, b) - \text{AUC}(S_{\text{base}}, a, b)}{\text{AUC}(S_{\text{self}}, a, b) - \text{AUC}(S_{\text{base}}, a, b)} - 1. \quad (7.2)$$

Figure 7.3 visualizes the calculation of the REU score. The REU score indicates the percentage decrease of AL self-selection sampling efficiency by AL foreign-selection relative to the baseline sampling scenario. If $\text{REU} = 0$, foreign- and self-selection are equally efficient and in the case of $\text{REU} > 0$, foreign-selection would be even better than self-selection. We deliberately defined the REU score so as to indicate *reduction* of sampling efficiency because we assumed that self-selection would constitute the upper efficiency bound for foreign-selection (cf. hypothesis H2 in Section 7.3.2.1).

A negative score with $-1 \ll \text{REU} < 0$ indicates that reusability is in evidence but comes with loss of sampling efficiency. We say that reusability is “high” for negative REU scores close to 0, and “low” for negative REU scores just slightly above -1 . A REU score as low or even below -1 indicates that reusability cannot be evidenced.

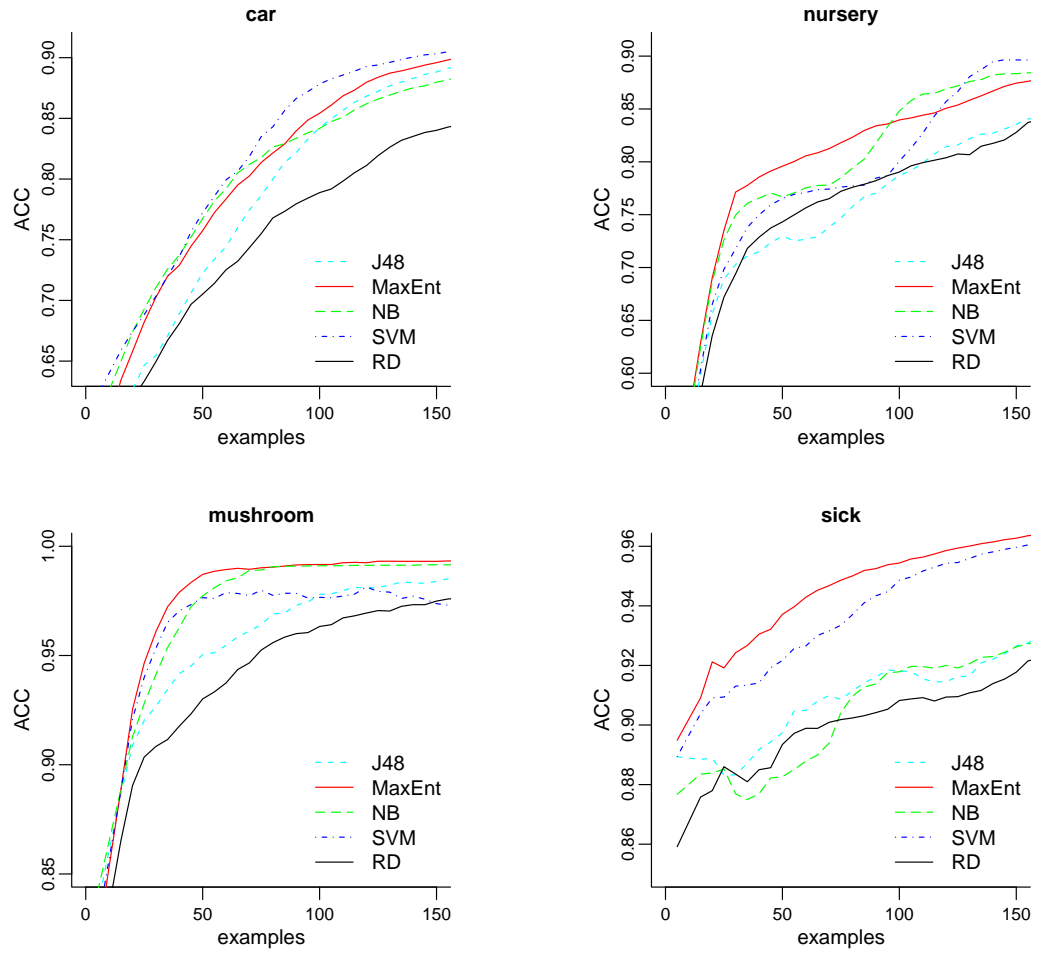


Figure 7.4: MaxEnt consumer with different selectors (as shown in legend) on UCI data sets.

7.3.1.2 Results

Sample Reuse on UCI Results of the experiments on sample reuse on the UCI data sets are shown in terms of REU and RAI scores in Table 7.2. These scores were calculated in the interval of [50; 150] examples to exclude the start-up phase and the convergence phase. While REU scores are indicators for reusability, RAI scores indicate the efficiency of AL self-selection. The RAI scores indicate that AL always exhibits a sampling efficiency above random sampling. However, sampling efficiency on the UCI data is on average relatively low, which holds especially for the SEGMENT and the SICK data sets and for the J48 learner. For visualization, learning

7.3 Empirical Investigation of Sample Reusability

CAR data set					
selector	REU score according to consumers				RAI score
	J48	MaxEnt	NB	SVM	
J48	0.00	-0.30	-0.47	-0.42	3.49
MaxEnt	-0.84	0.00	-1.04	-0.27	7.74
NB	-0.84	-0.11	0.00	-0.28	5.51
SVM	-0.90	0.26	-0.86	0.00	8.66
MUSHROOM data set					
selector	REU score according to consumers				RAI score
	J48	MaxEnt	NB	SVM	
J48	0.00	-0.59	-0.69	-0.19	1.42
MaxEnt	-0.05	0.00	-0.47	0.12	3.34
NB	0.00	-0.07	0.00	-0.02	8.14
SVM	-0.91	-0.43	-1.17	0.00	2.00
NURSERY data set					
selector	REU score according to consumers				RAI score
	J48	MaxEnt	NB	SVM	
J48	0.00	-1.13	-2.13	-0.93	1.92
MaxEnt	-0.33	0.00	-1.46	-0.07	6.13
NB	0.48	-0.09	0.00	0.14	4.30
SVM	-0.36	-0.35	-1.19	0.00	2.91
SEGMENT data set					
selector	REU score according to consumers				RAI score
	J48	MaxEnt	NB	SVM	
J48	0.00	-0.72	-0.47	-0.95	2.49
MaxEnt	-0.95	0.00	-1.24	-3.07	1.16
NB	-2.56	-2.04	0.00	-1.77	4.26
SVM	-4.35	-3.53	-2.39	0.00	1.38
SICK data set					
selector	REU score according to consumers				RAI score
	J48	MaxEnt	NB	SVM	
J48	0.00	-0.83	-0.54	-0.90	1.49
MaxEnt	0.26	0.00	-0.43	-0.77	5.20
NB	0.31	-0.90	0.00	-2.18	2.20
SVM	0.34	-0.18	-0.25	0.00	3.49

Table 7.2: REU and RAI scores on UCI data sets. Colors: $REU \geq 0$ and $REU \leq -1$

MUC7 corpus						
selector	REU score according to consumers					RAI score
	NB	HMM	MaxEnt	SVM	CRF	
NB	0.00	0.07	-0.19	0.13	-0.15	18.79
HMM	-0.48	0.00	-0.40	-0.29	-0.39	6.88
MaxEnt	-0.39	-0.05	0.00	0.12	-0.12	10.67
SVM	-0.40	-0.07	-0.20	0.00	-0.24	6.53
CRF	-0.38	0.01	0.02	0.05	0.00	6.73

PBGENE corpus						
selector	REU score according to consumers					RAI score
	NB	HMM	MaxEnt	SVM	CRF	
NB	0.00	-0.02	-0.01	-0.17	-0.13	5.03
HMM	-1.51	0.00	-0.29	-0.38	-0.35	6.65
MaxEnt	-3.47	-0.57	0.00	-0.08	-0.09	7.58
SVM	-2.22	-0.24	-0.22	0.00	-0.25	5.55
CRF	-3.58	-0.58	-0.06	-0.36	0.00	9.15

Table 7.3: REU and RAI scores on the NER scenarios. Colors: $REU \geq 0$ and $REU \leq -1$

curves for the MaxEnt consumer are shown in Figure 7.4.

On the UCI data sets there are many cases of good reusability. But, on the other hand, frequently a sample is not reusable in AL foreign-selection scenarios and REU scores drop considerably below the random selection baseline. Moreover, there are cases where only tiny amounts of sampling efficiency are sacrificed by AL foreign-selection (e.g., NB selector for a SVM consumer on the MUSHROOM data set).

Sample Reuse on NER Results on the NER data sets are shown in Table 7.3. The interval of interest for the computation of the REU and RAI scores is set to [10000; 30000] to be consistent with Chapter 4. RAI scores show that sampling efficiency of AL self-selection for any of the learners is on average higher than on the UCI data sets. Reusability for NER is given in all AL foreign-selection scenarios except for the NB-based consumer on the PBGENE corpus. Additionally, REU scores indicate that on a continuous scale, reusability is in most cases very high and less than 50 % sampling efficiency are forfeited. Overall, REU scores on the MUC7 corpus are generally higher than on the PBGENE corpus. For visualization, learning curves for the MUC7 corpus are shown in Figure 7.5.

In previous experiments on sample reuse, we found comparable results with QbC-based AL, while the current experiments are based on US (Tomanek et al., 2007a).

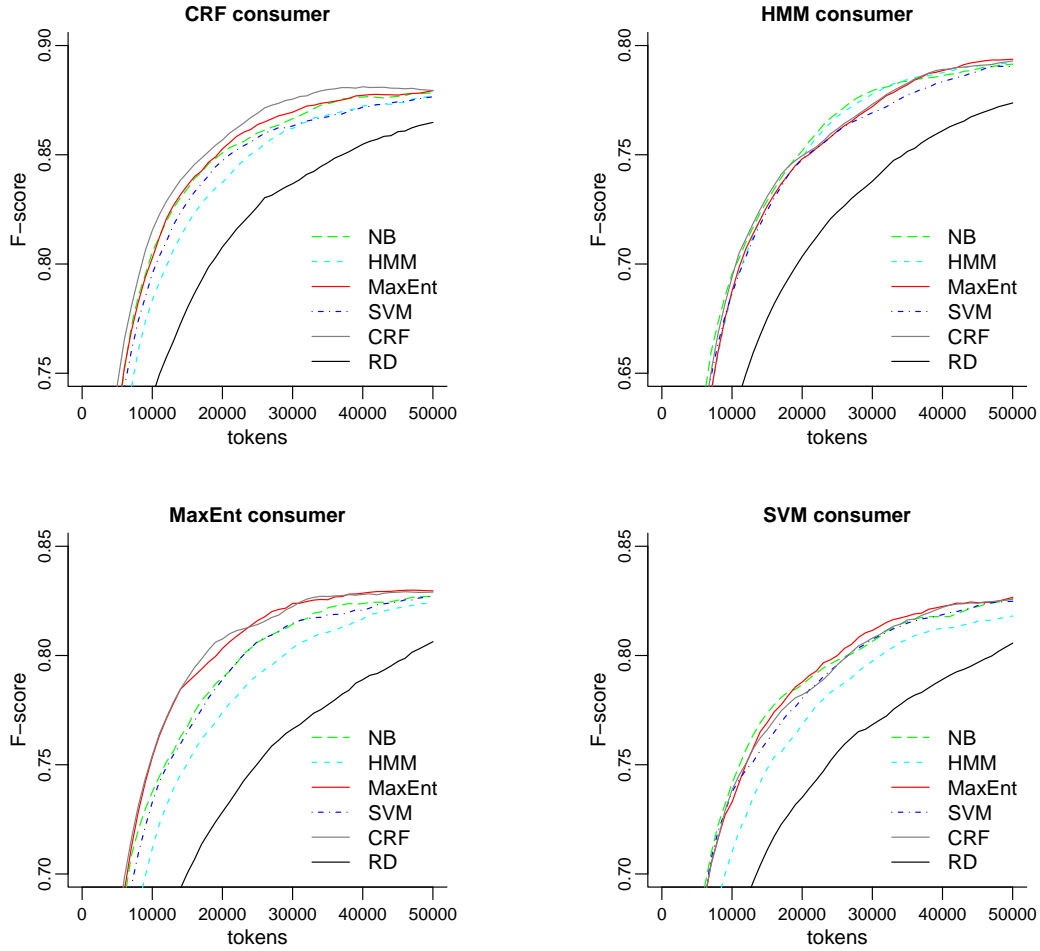


Figure 7.5: Sample reuse with different consumers and selectors in the NER scenario on MUC7 (selectors shown in legend).

Differences between the experiments on the UCI and the NER data sets lie in the way NER is evaluated (F-score vs. ACC) and the selection granularity (sentences vs. single examples). We performed additional experiments to test whether these differences explain the generally higher REU scores in the NER scenario.

Accordingly, we additionally evaluated the experiments on the NER data sets in terms of accuracy. However, we could not find a change of the overall picture of reusability. As an example, Figure 7.6 contrast F-score and accuracy evaluation of the MaxEnt consumer on the MUC7 corpus. As for selection granularity, the co-selection effect (cf. Chapter 4) might have a positive influence on reusability in the NER scenario. Experiments with a token-level selection granularity in the NER

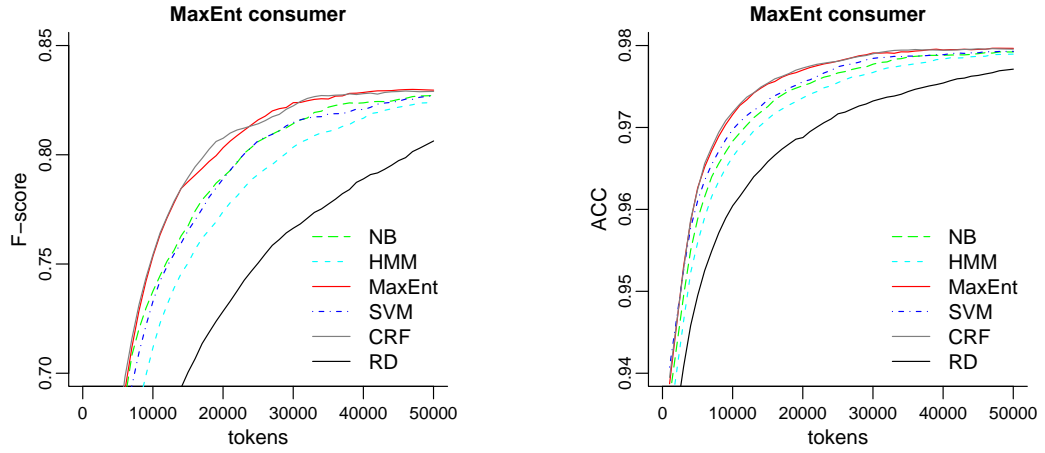


Figure 7.6: Evaluation of AL for NER with F-score and ACC.

setting disprove this suspicion and the same trends with respect to reusability can be observed. Figure 7.7 shows the foreign-selection learning curves for NER with token selection and the MaxEnt consumer. The same findings hold on the PBGENE corpus and the other consumers we tested.¹¹

7.3.2 Discussion of Hypotheses

7.3.2.1 Basic Hypotheses

H1: Limited Sample Reusability On the UCI data sets, one third of all AL foreign-selection scenarios exhibit REU scores ≤ -1 .¹² In these foreign-selection scenarios, sample efficiency considerably drops below that of random selection, often leading to extremely poor consumer performance as in the case of AL with a NB-based selector and an SVM-based consumer on the SICK data set.

As for NER, reusability can be recorded for all AL foreign-selection scenarios and REU scores rarely fall below -0.5 , indicating a high degree of reusability for this

¹¹According to these curves, NER with token selection has a better sampling complexity. However, annotation of isolated tokens is an unrealistic setting in practise (cf. Chapter 4). For the purpose of this experiment, token-level selection was enabled by simply splitting each sentence into the single tokens and running the same AL framework on these artificially down-scaled sentences.

¹²These are 15 out of the 60 cases of foreign-selection (highlighted in dark gray in Table 7.2).

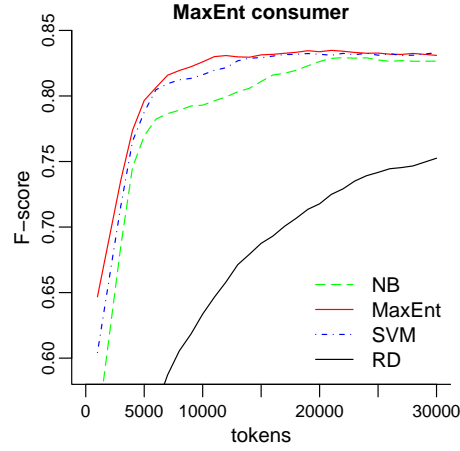


Figure 7.7: AL for NER in a token-selection scenario.

special learning problem. The only exception is that of foreign-selection for a NB-based consumer on the PBGENE corpus.¹³

While we did not find sample reusability to be generally given, we could still observe reusability in several cases. This includes for example the MUSHROOM data set and the NER scenario where reusability is extremely likely. According to these findings, we reject the strictly formulated hypothesis on generally limited sample reusability. At the same time, however, it has to be reported that reusability is consistently problematic for certain learning problems and data sets.

H2: Self-Selection as Upper Bound According to Tables 7.2 and 7.3, self-selection sampling efficiency is occasionally outperformed in foreign-selection scenarios. As for NER, this is the case in 6 out of the 40 foreign-selection scenarios; on the UCI data sets, 8 out of the 60 foreign-selection scenarios exceed the assumed upper bound.¹⁴ As an example, consider the combination of an SVM selector and a MaxEnt consumer on the CAR data set.

The observation that self-selection does not necessarily constitute the upper bound for sampling efficiency is remarkable: while self-selection aims to find the “optimal”

¹³For real-world applications one would avoid the application of a NB learner to the NER tasks: the NB assumption on feature independence is likely to be violated by the extremely rich set of features. For the experiments on sample reuse, we deliberately applied the NB learner to test limits of reusability in an extreme setting.

¹⁴Cases where REU > 0 are highlighted in light gray.

examples for a given learner T_1 , there are apparently scenarios where a learner T_2 estimates the utility of an example for learner T_1 more appropriately than T_1 itself.

In future work it should be tested whether self-selection constitutes the upper bound when approaches to statistically optimal AL instead of heuristics are applied.

H3: Selector-Consumer Pairings exhibit General Reusability Characteristics REU scores in Tables 7.2 and 7.3 contradict the assumption that there are pairings of learning algorithms for which general reusability characteristics hold. This holds especially for the UCI data sets where inconsistent reusability characteristics for the selector-consumer pairings have to be ascertained over the five data sets. Figure 7.4 on page 118 shows learning curves for the MaxEnt consumer and different selectors. Each plot constitutes its own reuse scenario and no characteristics hold on all plots. NB, for example, is a good selector for the MaxEnt consumer on the MUSHROOM data set, but performs poorly on the SICK data set for the MaxEnt consumer.

The situation is slightly different for the NER scenario and it appears that there indeed are tuples of learners which for this learning problem constitute pairings of consistent quality, such as a MaxEnt selector together with a CRF-based consumer. As another example, consider the combination of NB selector and MaxEnt consumer. Reusability is given for this combination on NER, but no consistent reusability characteristic can be observed on the UCI data set.

The only general pattern we found is that, in the NER scenario, in most foreign-selection scenarios with a NB-based consumer REU scores are low and $\text{REU} > 0$ could not be evidenced. In contrary, the NB learner is commonly a good selector in foreign-selection scenarios. In conclusion, foreign-selection for a NB consumer should preferably be avoided.

In contrast to our original assumption, we conclude that reusability depends highly on the particular data set and it is not possible to generalize which combinations of learners generally work well together. Our findings also disagree with the assumption formulated by Vlachos (2004) that the stronger learner should be used as selector to increase reusability. The opposite is often the case as evidenced by the combination of NB and CRF learners on the NER data sets.

Given the findings that reusability is not generally limited and that no selector-consumer pairings with general reusability characteristics could be found, we turn here to a set of more specific follow-up assumptions on explanatory factors for reusability.

7.3.2.2 Follow-Up Hypotheses

H4: Relatedness of Models Explains Reusability In line with Baldridge and Osborne (2004), we assume that sample reusability depends on the degree of relatedness between selector and consumer. It should be noted, that by selector and consumer we here explicitly refer to the *models* induced by the respective learners given a learning problem and data set.

Approaches to quantifying the relatedness of models have been studied intensively in context of ensemble learning and multi-classifier systems (Ho et al., 1994; Kuncheva and Whitaker, 2003). Relatedness is usually measured by the degree of correlation between the *predictions* of models. In the context of AL, we say that two models are related when they lead to a highly correlated utility ranking of unlabeled examples. To quantify relatedness, we follow Baldridge and Osborne (2004) and apply Spearman’s rank correlation coefficient ρ to the utility rankings of both models.

The assumption is that high relatedness scores for combinations of two models come with high REU scores, while low relatedness scores imply low REU scores. We test this assumption on the NER data. All NER learners are trained on a random subsample of 10,000 tokens.¹⁵ Utility rankings of the examples in the test set are then compared for all tuples of models. Table 7.4 shows the relatedness scores obtained by Spearman’s rank correlation on the MUC7 and the PBGENE corpus.

By definition, relatedness scores are symmetrical. However, reusability, according to Tables 7.2 and 7.3, is not. When the learners of selector and consumer are exchanged, reuse scores differ. This is, for instance, evidenced by the tuple of SVM and MaxEnt learner for which we obtained a high relatedness score of 0.81 on the MUC7 corpus. An AL sample obtained by a MaxEnt-based selector is perfectly reusable by an SVM-based consumer with sampling efficiency above that of SVM self-selection with $\text{REU} = 0.12$. However, when SVM is used to select for a MaxEnt consumer, reusability drops to $\text{REU} = -0.20$. More extremely, this is also the case for a tuple of NB and CRF on the PBGENE corpus. Though strongly related (0.69), CRF as a selector for NB fails miserably with $\text{REU} = -3.58$. The other way round, however, high reusability with $\text{REU} = -0.13$ is obtained. Thus, a high relatedness score does not necessarily imply high reusability.

The very good reusability of a sample obtained by AL with a NB selector for a HMM consumer is in contrast to the rather low relatedness score for HMM and NB of 0.47 on the MUC7 corpus. A low relatedness score does thus not necessarily imply a low level of reusability. While a high rank correlation coefficient often accompanies reusability (as for the MaxEnt-CRF tuple), one cannot conclude the

¹⁵We also tested random samples of different sizes but did not obtain essentially different results.

MUC7 corpus					
	NB	HMM	MaxEnt	SVM	CRF
NB	1	0.47	0.68	0.54	0.68
HMM	0.47	1	0.59	0.46	0.60
MaxEnt	0.68	0.59	1	0.81	0.94
SVM	0.54	0.45	0.81	1	0.74
CRF	0.68	0.60	0.94	0.74	1

PBGENE corpus					
	NB	HMM	MaxEnt	SVM	CRF
NB	1	0.47	0.69	0.43	0.69
HMM	0.47	1	0.57	0.35	0.57
MaxEnt	0.69	0.57	1	0.69	0.92
SVM	0.43	0.35	0.69	1	0.59
CRF	0.69	0.57	0.92	0.59	1

Table 7.4: Relatedness scores based on Spearman’s rank correlation coefficient on unlabeled examples for any combination of two learners.

opposite from low correlation coefficients. This emphasizes that different samples can also lead to similar model performances and anticipates an issue that we will address in the following sections: a sample that is highly divergent from the self-selection sample is not necessarily inferior in terms of classifier performance achieved by the consumer.

H5: Similarity of Samples explains Reusability Although the analysis of H4 already gave hints for the opposite being true, an obvious assumption would be that similarity of samples obtained in self- and foreign-selection mode is a relevant factor for reusability. This assumption is based on the intuition that different selectors may select from other parts of the instance space and the more the covered space of a foreign-selector diverges from that of the self-selector, the lower the REU scores would be. Under this assumption, a situation with $\text{REU} \leq -1$ could be interpreted as a scenario where the AL sample does not cover the relevant areas for the consuming learner to build appropriate hypotheses. For another learner, however, according to different model assumptions, the same sample may still be adequate for finding a good model.

Instead of comparing the samples on the example level, we compare how samples distribute over the input space. Therefore, the input space, represented by the set of all unlabeled examples in the pool \mathcal{P} , is clustered. Here we apply agglomerative

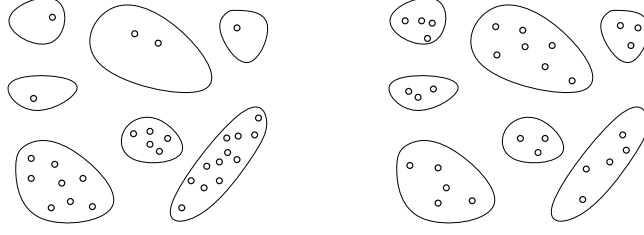


Figure 7.8: Distribution of samples S_1 (left) and S_2 (right) over common clustering.

hierarchical clustering. The distance between two clusters is calculated according to the average linkage method based on the Euclidean distance (Everitt et al., 2001). The hierarchical clustering is flattened down to $k = 20$ clusters. The examples in a sample S are then assigned to this clustering according to each example’s proximity to the cluster centroid (Everitt et al., 2001).

D_S represents the distributions of the examples of sample S over the clustered input space. This distribution gives the percentage of a sample’s examples falling in each cluster. Figure 7.8 visualizes this for two samples S_1 and S_2 obtained by two different selectors. S_1 covers the lower clusters more densely, while examples of S_2 are almost equally distributed over all clusters. The similarity of the two samples S_1 and S_2 is estimated based on the divergence of their distributions D_{S_1} and D_{S_2} by means of the KLM metric (Equation 3.15 on page 40). The KLM score ranges in the interval of $[0, 1]$ and the lower the KLM scores, the more similar two samples are with respect to their distribution in the clustered space. We calculate this similarity by $\text{SIM}(S_1, S_2) = 1 - \text{KLM}(D_{S_1}, D_{S_2})$. In the above example, a similarity of $\text{SIM}(S_1, S_2) = 0.48$ is obtained.

Our hypothesis that distributional similarity of samples is a relevant factor for reusability is operationalized by the assumption that similarity (SIM) correlates with reusability (REU). The intuition is that high sampling efficiency of foreign-selection comes with high similarity of the samples. We tested this hypothesis only on the UCI data sets.¹⁶ SIM and REU scores were calculated for samples of 100 examples.¹⁷ Correlation between REU and SIM scores was calculated for each data set separately omitting REU and SIM scores of the self-selection scenarios.

Table 7.5 shows correlation coefficients between the REU and SIM scores. Pearson’s correlation coefficients range between 0.03 and 0.37, which indicates a comparatively low (linear) relationship. Spearman’s correlation coefficients are also very low on

¹⁶Due to high computational complexity of clustering as a result of large feature spaces, we refrained from testing the clustering on the NER corpora.

¹⁷Since REU scores are defined over a range of examples, we here chose the interval $[99, 101]$.

	CAR	MUSHROOM	NURSERY	SEGMENT	SICK
Pearson's ρ	0.30	0.24	0.03	0.37	0.23
Spearman's ρ	0.29	0.19	0.08	0.40	0.14

Table 7.5: Pearson's and Spearman's correlation coefficients on REU and SIM scores.

average ranging from 0.08 to 0.4. This shows that SIM scores are also not suitable for ranking the foreign-selection scenarios according to their REU scores.

To illustrate this finding with examples, Tables 7.6 shows the SIM scores on the MUSHROOM data set. According to these scores, all samples obtained by AL selection diverge from the unbiased distribution of this data set. The unbiased distribution is the distribution of the complete pool \mathcal{P} . Samples obtained with AL and a NB- or SVM-based selector exhibit a lower similarity to the unbiased distribution than do those based on a J48 or a MaxEnt selector.

Table 7.7 shows REU scores on the MUSHROOM data set. Once again, SIM scores are symmetrical. While $\text{SIM}(S_{\text{SVM}}, S_{\text{NB}}) = 0.85$ signals that the samples obtained from AL with a SVM selector and with a NB selector diverge, a sample S_{NB} exhibits a high sampling efficiency for the SVM consumer ($\text{REU} = 0.05$). In the reverse situation, a sample S_{SVM} is not appropriate for the NB consumer ($\text{REU} = -1.24$). Furthermore, consider a sample S_{SVM} for a MaxEnt or a NB consumer. Distributional similarity is very similar, $\text{SIM}(S_{\text{SVM}}, S_{\text{MaxEnt}}) = 0.86$ vs. $\text{SIM}(S_{\text{SVM}}, S_{\text{NB}}) = 0.85$, but reusability of S_{SVM} for a MaxEnt consumer is much better than for a NB consumer ($\text{REU} = -0.53$ vs. $\text{REU} = -1.24$). REU and SIM scores of all UCI data sets are shown in Tables D.1 and D.2 (Appendix, pages 229 and 230).

These experiments show that the distributional similarity of samples is also unable sufficiently to explain reusability. Once again, SIM scores are symmetrical which is problematic as discussed in the context of H4. Another problem with measuring distributional similarity based on clustering is that, for clustering, all dimensions of the input space are considered equally important. Thus, two sample distributions that differ highly in many unimportant dimensions but are very similar in the few really important dimensions, would still get a low SIM score. This, however, does not reflect the situation of a learner that actually weights the dimensions.

H6: Influence of Sample Bias on Classifier Training As a follow-up to H5 we hypothesize that sensitivity to sample bias is a relevant factor for reusability characteristics. This sensitivity, we assume, is expressed by the fact that a model learned on

MUSHROOM data set					
selector	J48	selector			unbiased sample
		MaxEnt	NB	SVM	
J48	1.00	0.88	0.80	0.80	0.90
MaxEnt	0.88	1.00	0.89	0.86	0.80
NB	0.80	0.89	1.00	0.85	0.67
SVM	0.80	0.86	0.85	1.00	0.71

Table 7.6: Mutual distributional similarity (SIM score) between the samples of 100 examples selected with AL using different selectors and distributional similarity between these samples and an unbiased sample as represented by the complete pool \mathcal{P} .

MUSHROOM data set					
selector	J48	consumer			SVM
		MaxEnt	NB	SVM	
J48	0.00	-0.48	-0.73	-0.17	
MaxEnt	0.06	0.00	-0.49	0.16	
NB	0.07	-0.02	0.00	0.05	
SVM	-1.14	-0.53	-1.24	0.00	

Table 7.7: REU scores on samples of 100 examples.

an adversarial sample exhibits highly different model parameters or feature weights compared to one learned on a self-selected sample.

To quantify the influence training data has on model learning, we determine the resulting feature rankings after training models on different samples. Feature rankings are obtained by a wrapper approach based on simple hill climbing (Kohavi and John, 1997). Subsequently, tuples of feature rankings are compared. A tuple always consists of the feature ranking obtained from a model learned on a foreign-selection sample and the feature ranking of the model learned on the self-selected sample.

Comparison of feature rankings is based on a weighted version of Spearman’s rank correlation coefficient.¹⁸ Accordingly, the feature ranking score $FR(S_{T_1}, S_{T_2})$ shows the correlation of the feature rankings of a model induced by learner T_2 on a foreign-selection sample from AL with a selector based on T_1 and a self-selection sample the selector was based on T_2 .

Hypothesis H6 is thus operationalized by the assumption that the FR scores correlate highly with the REU scores in the foreign-selection scenarios. However, our

¹⁸Refer to Appendix D.1 for details on the weighted rank correlation.

	CAR	MUSHROOM	NURSERY	SEGMENT	SICK
Pearson's ρ	0.16	-0.35	0.46	-0.38	-0.06
Spearman's ρ	0.04	-0.49	0.45	-0.42	-0.21

Table 7.8: Pearson's and Spearman's correlation coefficients on REU and FR scores.

MUSHROOM data set				
selector	consumer			
	J48	MaxEnt	NB	SVM
J48	1.00	0.32	0.34	0.41
MaxEnt	0.79	1.00	0.11	0.51
NB	0.70	0.40	1.00	0.47
SVM	0.77	0.38	0.14	1.00

Table 7.9: Similarity of feature ranking (FR score): feature rankings of a consumer trained on a self-selected sample and a foreign-selected sample are compared. Samples have 100 examples. As an example, $FR(J48, MaxEnt) = 0.32$ means that the feature ranking of a MaxEnt consumer trained on a sample selected by a MaxEnt selector has a weighted Spearman's rank correlation coefficient to the feature ranking of a MaxEnt consumer trained on a sample selected by a J48 of 0.32.

experiments disprove this assumption. Table 7.8 shows the correlation coefficients for REU and FR scores on the UCI data sets. With the exception of the NURSERY data set, correlation coefficients are low or even negative.

As an example, compare the REU scores of Tables 7.7 with the FR scores of Table 7.9 for the MUSHROOM data set. While a J48 consumer has FR scores between 0.7 and 0.79 for the different selectors, REU scores are more diverse. While SVM is a miserable selector for J48 ($REU = -1.14$), MaxEnt and NB exhibit a higher sampling efficiency for J48 than does J48 self-selection ($REU = 0.06$ and 0.07). While correlation coefficients on the NURSERY data set are a bit higher (Pearson's $\rho = 0.45$), examination of the REU and FR scores reveals problematic cases. For the J48 consumer, REU scores vary between -0.21 and 0.66 . But FR scores, in contrast, are all nearly identical (around 0.5) for the different selectors (REU and FR scores for all data sets can be found in the Appendix in Tables D.1 and D.3 on pages 229 and 231).

Overall, the FR score is inadequate either for predicting the REU score and for ranking the selectors according to their appropriateness for a particular consumer. A twisted feature ranking may still lead to a model with similar accuracy compared to a model which is induced from a self-selected sample. This is, for example, the case

on the SICK data set for the MaxEnt consumer. An SVM-selected sample exhibits a low FR score of 0.22, but the REU score of -0.12 is high, in comparison.

Reusability cannot be explained by the fact that models learned on different samples exhibit similar feature rankings. In consequence, a foreign-selection sample from which a learner induces a model θ , that is highly disparate from a model θ' induced by the same learner but from a self-selection sample, may still perform similarly well or even better than θ' .

H7: Class Distribution affects Reusability In all previous experiments, we could not find factors that generally explained reusability independent of the learning problem and data set. The only distinct pattern found is that reusability is comparatively high on the NER task and REU scores ≤ -1 are evidenced only in the unrealistic setting with a NB consumer. What is more, the NER task also appears highly appropriate for AL itself, as evidenced by comparatively high RAI scores. Accordingly, a question of high relevance concerns the special characteristics of the NER task as a learning problem, as distinct from the classifications problems found in the UCI data sets.

An outstanding characteristic is clearly the class distribution. The NER task is subject to a considerable class imbalance (cf. Table 4.2) with the OUTSIDE class covering about 89% of all tokens in the MUC7 corpus and 93.5% in the PBGENE corpus. In the following, we test whether such a distinctive class imbalance is a relevant characteristic for reusability. Our hypothesis thus is that high REU scores are obtained when such a class imbalance is given and REU scores are low for uniform class distribution.

For the experiments, UCI data sets were re-sampled so that their class distributions resemble those of the NER task: one class is the majority class and the other classes are approximately equally distributed.¹⁹ The class distribution of the original and the re-sampled data sets is shown in Table D.4 (Appendix, page 232).

Figure 7.9 exemplarily contrasts the learning curves on the original and the re-sampled data sets for the MaxEnt consumer. On the SEGMENT data set, AL sampling efficiency and reusability are improved by re-sampling. On the CAR data set, reusability is improved for the NB selector, but, the J48 selector now drops down to random selection sampling efficiency. On the NURSERY data set, however, re-sampling causes NB and SVM selectors to fail miserably in selecting for the MaxEnt

¹⁹Such re-sampling was only possible for the three largest UCI data sets: CAR, NURSERY, and SEGMENT data sets. The other UCI data sets had a prohibitively small number of examples. More details on re-sampling procedure is given in Appendix D.3.

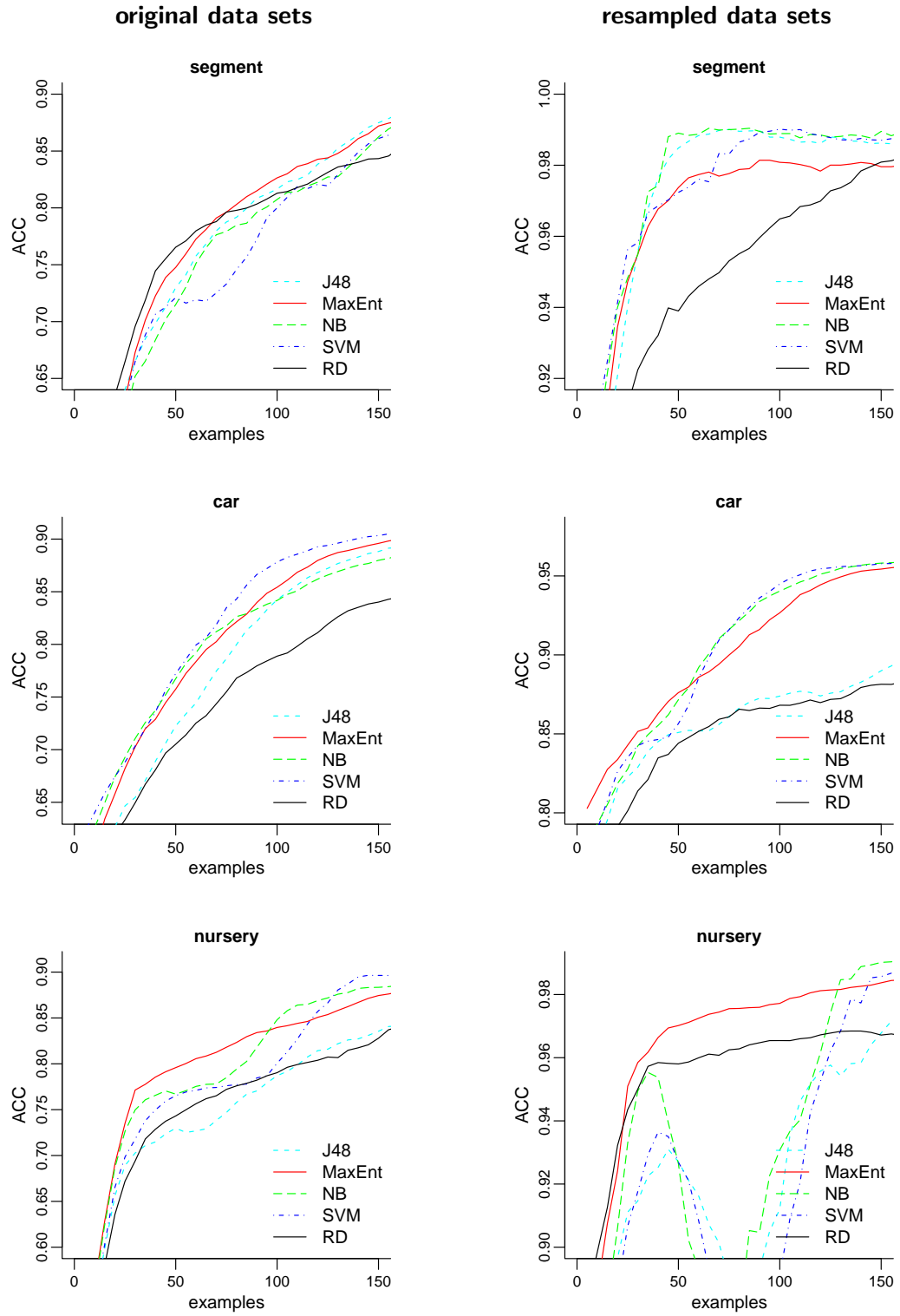


Figure 7.9: MaxEnt consumer on re-sampled and original UCI sets (legend shows selectors).

consumer. Detailed REU and RAI scores can be found in Table D.5 (Appendix, page 232).

Our experiments show that re-sampling helped especially on the SEGMENT data set which, in its original version, has seven uniformly distributed classes. However, uniformly distributed classes are not necessarily an impediment to sample reuse: On the MUSHROOM data sets, the two classes are also uniformly distributed (52 % vs. 48 %), but reusability is still mostly in evidence. Moreover, preliminary AL experiments on re-sampled NER data showed that, even when all classes are equally frequent, AL sampling efficiency and reusability are still good.

7.3.3 Conclusions

In this section we have empirically studied the case of sample reuse and reusability on general classification problems as found in the UCI repository and on the NER task, a special class of learning problems of its own. Our investigation into characteristics of and explanatory factors for reusability was driven by three initial hypotheses and four follow-up hypotheses.

To start with (H1), we assumed that sample reusability would be a very unlikely scenario and thus empirically one would find only few cases where $\text{REU} \gg -1$. While we indeed did find many cases where $\text{REU} \leq -1$, in more than 50 % of all cases sample reuse resulted in a sampling efficiency better than that of random selection.

The hypothesis that self-selection would pose the upper bound for sampling efficiency in sample reuse scenarios (H2) was rejected. In contrast, we observed cases where AL foreign-selection clearly outperformed self-selection sampling efficiency.

The next hypothesis (H3) stated that selector-consumer pairings would always be subject to the same reusability characteristics. On the contrary, we could not identify such a pattern but found that the appropriateness of such pairings for sample reuse depends highly on the specific data set.

Given this finding, in a follow-up hypothesis (H4) we assumed that the relatedness of models induced by different learners is an explanatory factor for reusability. Again, experiments disproved this hypothesis – for highly unrelated selector-consumer pairings we also found good reusability. Similarly, another follow-up hypothesis (H5), stating that distributional similarity of resulting samples explains reusability, was disproved. Instead, we found that distributionally dissimilar samples can lead to models of comparable performance.

As another follow-up (H6), we studied the influence of sampling bias induced by sample reuse on the learned models. This was operationalized by the feature ranking a model would produce on different samples. We assumed similar feature rankings to be an indicator for the presence of reusability and disparate rankings to be an explanation for situations where $\text{REU} \leq -1$. We found that even models of disparate feature rankings may exhibit comparable performance.

Finally, since all hypotheses previously were disproved, we asked about the important characteristics of the NER scenario resulting in generally good reusability characteristics. We assumed class distribution (H7) to be such a characteristic and a supporting factor of reusability. While this was partly confirmed by our experiments, we also here found cases of reusability that are in contrast to H7.

Overall, we discovered that sample reuse on the NER tasks in all realistic settings works very well. This, however, does not hold in general for arbitrary learning problems. Our approaches to find general factors causing or preventing reusability failed. We thus conclude that reusability crucially depends on the data set at hand and the characteristics of the specific learning problem. This brings us back to an issue addressed before in the context of learning under sample selection bias and covariate shift. This is that the sensitivity of learners on sample bias depends on the combination of learning algorithm, data set, and learning problem (Fan et al., 2005). Future work should study methods to quantify the effect of sample selection bias, given a learner and an arbitrary data set.

Focusing on the special problem of reusability in the context of AL, this constitutes a difficult and specific aspect of the issue of quantifying sensitivity to sample bias because (a) the question here is whether sensitivity affects reusability, and (b) in the AL scenario we only have access to unlabeled data.

7.4 Boosting Sample Reusability in the Agnostic Setting

In the previous section of this chapter we found that reusability should not be blindly assumed for an arbitrary learning task but that it is likely in the NER scenario. However with reference to the NER scenario, foreign-selection often exhibits considerably lower sample efficiency than self-selection. As an example, consider the case of an HMM-based selector in combination with a CRF-based consumer on the MUC7 corpus ($\text{REU} = -0.39$, cf. Table 7.3 on page 120).

According to our findings on the MUC7 and the PBGENE corpus, in a scenario for which it is decided that the final consumer is to be CRF-based, one would thus rather choose a MaxEnt-based selector than a HMM-based one. However, as argued

at the beginning of this chapter, for many real-life annotation scenarios, the final consumer for which labeled training data has to be generated is not known before or during the data acquisition time. In consequence, it is unclear which learner to apply during selection in order to sacrifice as little sampling efficiency as possible.

We propose here an approach to AL which aims to generate a sample likely to be highly reusable for typical learners for a specific learning problem. This means that we aim to find examples that are good on average – although presumably not perfect – for possible consumers. A core assumption for this approach is that there are indeed examples which are useful to some extent for all learners. If this is not the case, reusability can only be assumed in a self-selection scenario.

7.4.1 Heterogeneous Committees

QbC-based AL is founded on the disagreement regarding different hypotheses. In the case of QbB, the disagreement with a committee of models trained on different subsets of the labeled training data \mathcal{L} is considered. It has been shown by Melville and Mooney (2004) that a more diverse committee is better suited to measuring the utility of an example appropriately for AL. However, approaches to building diverse committees are generally based on a single learning algorithm.

Our approach to generating a sample that is likely to be highly reusable by different learners is based on the idea of a *heterogeneous* committee built from models induced by different learners. As in the QbB setting, each model is trained on a random subsample of \mathcal{L} , but each model is based on another learner.

The assumption is that such a *heterogeneous committee* (HEC) rates as generally useful such examples that are on average well-suited for most committee members. For a special learning problem that is already well-understood, there may be only a small number of learners that lead to completely different models. When all such approaches are melded together in a HEC, chances are high that a final consumer will have similarities to the committee, so that foreign-selection based on a HEC on average results in better reusability than the worst-case foreign-selection scenario.

7.4.2 Experiments and Results

We tested the HEC to improve reusability for an unknown consumer on the NER problem. The NER problem is indeed well-studied; most approaches to NER exhibit overlapping feature sets. For a realistic scenario we generated a HEC that did not contain the final consumer. The HEC was populated with the following five freely available tools for ML-based NER:

selector	Stanford CRF consumer	
	MUC7	PBGENE
Mallet CRF	-0.08	-0.13
Mallet MaxEnt	-0.24	-0.21
Liblinear SVM	-0.21	-0.35
OpenNLP MaxEnt	-0.22	-0.32
Lingpipe HMM	-0.39	-0.38
heterogeneous committee	-0.18	-0.14

Table 7.10: REU scores for Stanford CRF consumer trained on samples obtained by AL foreign-selection with different selectors. The heterogeneous committee comprises all of these selectors. **Best** and **worst** performing foreign selectors are highlighted.

- An NE tagger based on MALLET’s implementation of a CRF, as well as two variants, one using a MaxEnt learner and the other a NB learner. These approaches have been used before in this chapter.
- An SVM-based tagger, also used for previous experiments this chapter.
- Lingpipe’s HMM implementation, also used for previous experiments.
- The well-known OpenNLP MaxEnt tagger (OpenNLP, 2008).

While the taggers based on MALLET and the SVM-based tagger employ the same features, Lingpipe’s HMM and OpenNLP’s MaxEnt tagger are based on their own distinct feature sets. Another important difference between the members of the HEC is that, when tested in a passive-learning scenario, the taggers yield quite different performance values between 77-88 % F-score on the MUC7 corpus and 75-83 % F-score on the PBGENE corpus. We ran AL with each single tagger as described in Section 7.3 as well as AL with a HEC. The Vote Entropy utility function was used for AL with a HEC. AL runs were evaluated with the Stanford CRF-based NE tagger (Finkel et al., 2005).²⁰

Figure 7.10 compares self-selection for the Stanford CRF consumer with all foreign-selection scenarios on the MUC7 and the PBGENE corpus. The direct juxtaposition of self- and foreign-selection performance shows that reusability is given for all scenarios. However, all five foreign-selectors fell behind self-selection sampling efficiency with MALLET CRF and MaxEnt being the best and OpenNLP’s MaxEnt/Lingpipe’s HMM the worst foreign-selection scenarios. The reduction of sampling efficiency in terms of REU scores is shown in Table 7.10.

²⁰Compared to our CRF-based NE tagger, the Stanford NE tagger performs about 2 percentage points better on most data sets tested.

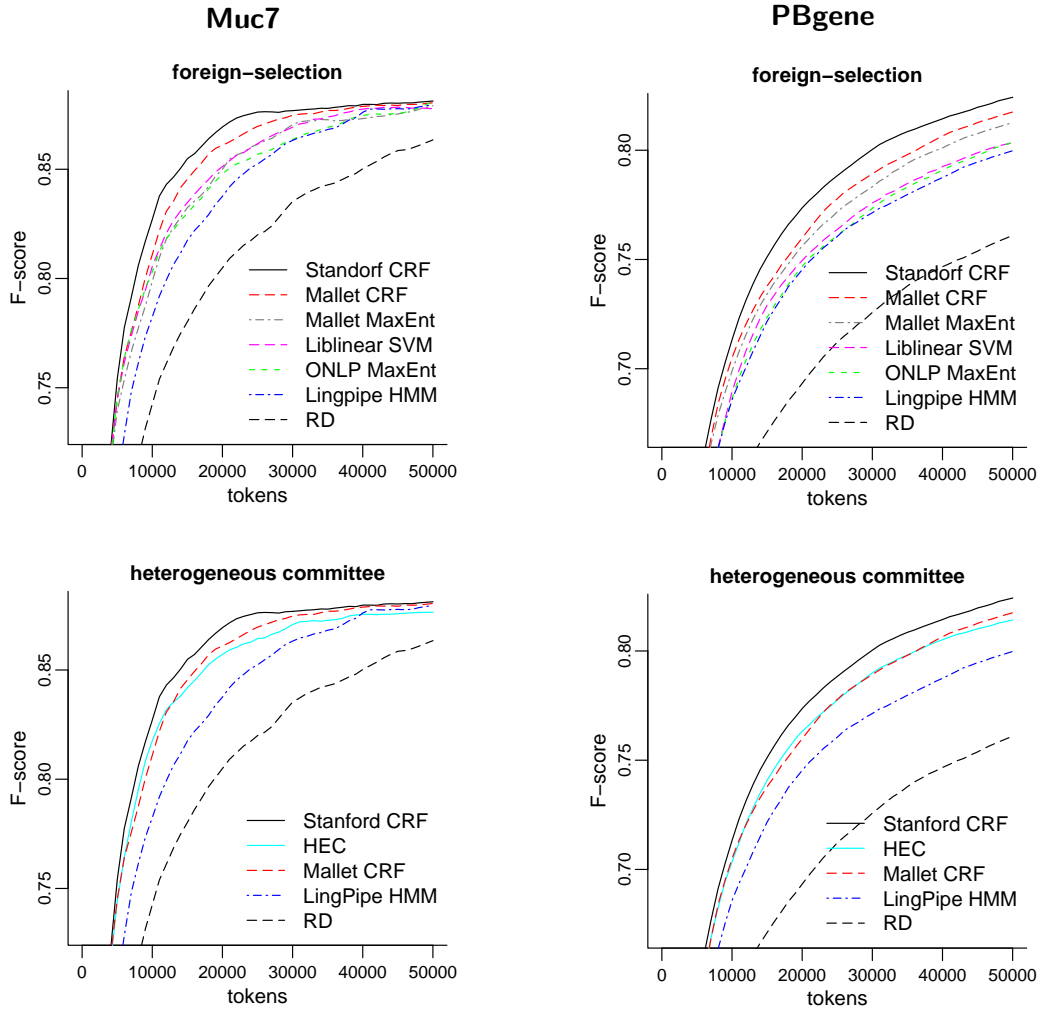


Figure 7.10: Real-world sample reuse scenario with self- and foreign-selection, and selection with a heterogeneous committee. Stanford CRF used as consumer, selectors indicated by the legend in the plots (ONLP refers to OpenNLP).

In Figure 7.10 performance of HEC, best-case and worst-case foreign-selection is contrasted with self-selection for Stanford CRF. HEC foreign-selection outperforms the second-best foreign-selector (MALLET MaxEnt) on the MUC7 corpus and is even on par with the best foreign-selector (MALLET CRF) on the PBGENE corpus. On both MUC7 and PBGENE, the worst-case foreign-selection performance is clearly exceeded. This shows that in a scenario where one does not know about the final consumer to be used, the application of HEC foreign-selection considerably reduces

the risk of choosing the wrong selector. In consequence, HEC foreign-selection considerably increases sample flexibility.

7.5 Summary and Conclusions

Sample reuse and reusability are two issues hardly addressed in studies on AL. However, in real-world scenarios, sample reuse is of high practical relevance. To the best of our knowledge, our work is the first to define clearly AL sample reuse. Moreover, it is the first large-scale empirical study on AL sample reuse. Our experiments showed that reusability is a challenging problem and that it is difficult to make general statements. Most importantly, it showed that the question of whether a foreign-selection sample is reusable by a particular learner, depends on the combination of learners, data set and learning problem.

While this chapter could not completely describe the true nature of AL sample reusability or find explanatory factors for successful sample reuse, it provides a starting point for further investigations which may build on the empirical study and the arrangement of reusability in the context of learning under sample selection bias. Future work in this direction should focus on the quantification of a learner's sensitivity to sample selection bias given a specific learning problem.

For the NER task, our experiments provide encouraging results as they indicate flexible sample reusability for this learning problem. Whatever selector is being used, the sample is still much more valuable for learning any of the considered consumers than is a random sample. Thus, in the NER scenario, Requirement 5 is met by our NER-specific AL framework. From a more practical point of view, this chapter showed that for the NER task, foreign-selection with a MaxEnt-based selector is well suited to acquiring labeled training data for a CRF-based consumer. Chapter 10 builds on this finding to speed up the AL sampling process considerably without sacrificing too much sampling efficiency.

Finally, an approach has been proposed to keep sampling efficiency high in a foreign-selection scenario where the final consumer is unknown. Experiments on foreign-selection with a heterogeneous committee provide first evidence that this approach is well-suited for reducing the risk of low reusability by a badly chosen learner for a foreign-selection scenario. Future work on this issue should extend our experiments to other consumers besides the Stanford CRF tagger and also apply approaches based on HECs to other learning problems.

Part III

Active Learning with Multiple Criteria

Chapter 8

Multi-Criteria Active Learning

In the previous part of this thesis, we implicitly assumed AL to be subject to a single selection criterion, i.e., the utility of an example p quantified by function $u(p, \theta)$. With utility we refer to the usefulness of an example for model training. Therefore, the example with the highest utility was considered the optimal solution of one AL iteration, as formalized in Equation 3.1 on page 31. In the case of batch-mode AL, the $|\mathcal{B}|$ examples with the top ranking utility scores would be selected.

The utility function was defined as consisting of two components, namely informativeness and representativeness (Definition 9 on page 31). Thus, we already tacitly anticipated that utility may be a function of two dependent variables. However, even with the possibility of incorporating representativeness, we consider the utility as a monolithic function resulting in a single selection criterion. As a matter of fact, the majority of approaches to AL employ a utility function based on informativeness alone. However, besides the mere utility of an example, real-world scenarios may require additional criteria – possibly conflicting with utility – to be considered during selection as well. Such criteria include, for instance, misclassification costs or *true* annotation effort in terms of time or money.

In the third part of this thesis, the assumption of a single criterion is relaxed leading to a generalization of our AL framework to one with multiple criteria for selection. This chapter formalizes AL with multiple criteria as a decision problem and presents two general methods that deal with multiple criteria in a selective sampling scenario. Finally, a brief overview of previous work on AL with multiple criteria is given.

The remaining chapters of part three present concrete instantiations of AL in real-world scenarios with multiple, mostly conflicting, selection criteria. While utility of an example is always the primary criterion, different secondary criteria are considered. In Chapter 9, the selected examples should also be useful for deviant learning problems, in Chapter 10, the secondary criteria aims at reducing class imbalance within the set of labeled examples, and Chapter 11 strives to find such examples both useful and economical in terms of annotation time.

8.1 Decision-Theoretic Formulation of Active Learning

AL with multiple criteria (MCAL) can be formalized as a problem of multi-criteria decision-making. Multi-criteria decision-making describes the process of selecting the best decision from the decision space given multiple criteria and some preference structure over these.

Multi-criteria decision-making is usually subdivided into *multi-objective* decision-making which applies to problems with a continuous decision space and *multi-attribute* decision-making for problems with discrete and finite decision spaces. According to this classification, multi-criteria AL falls into the second class because the decision space is represented by the set of unlabeled examples.

Following Keeney and Raiffa (1976), we define a *multi-attribute decision-making* (MADM) problem by a set of feasible alternatives d from the alternative space \mathcal{D} . An alternative is described by multiple attributes a . An *attribute-value function* $v_a(a_d)$ evaluates the value of attribute a of an alternative d .

Moreover, it is assumed that there is an order relation \succeq operating on all q attribute-value functions $v_a(a_d)$ for d so that an alternative d is mapped to an ordered space (\mathbb{R}^q, \succeq) . An alternative d' is said to *dominate* (or be superior to) an alternative d'' , if $\forall a = 1, \dots, q : v_a(a_{d'}) \geq v_a(a_{d''})$ and $\exists a$ for which $v_a(a_{d'}) > v_a(a_{d''})$. MADM problems usually have a set of non-dominated alternatives, where each one represents a particular trade-off between the considered attributes.

Transferring this general definition to the MCAL scenario, an alternative d refers to an unlabeled example p with the alternative space being the set \mathcal{P} of unlabeled examples from which AL selects. An example can have several attributes, which we call *criteria* in the context of AL, such as the utility $u(p, \theta)$ or the annotation time $t(p)$. Each such criterion is mapped to a value in \mathbb{R} by an attribute-value function $v_a(p)$ which can be arbitrarily complex. As a simple example, consider the attribute-value function for annotation cost

$$v_t(t(p)) = \frac{t(p) - \min_{p' \in \mathcal{P}} t(p')}{\max_{p' \in \mathcal{P}} t(p') - \min_{p' \in \mathcal{P}} t(p')}$$

which does nothing more than normalize $t(p)$ to the interval $[0, 1]$.

We define a vector $\vec{v}(p) = (v_1(p), \dots, v_q(p))$ to hold all attribute-value functions for an example p .

Considering the general AL framework (Algorithm 1 on page 32) and the NER-specific, batch-mode AL framework (Algorithm 2 on page 55), MCAL leads to modifications of Step 2 in both algorithms, only. In Algorithm 1, the optimization

changes to

$$p^* = \operatorname{arg}\mathbf{max}_{p \in \mathcal{P}} \vec{v}(p)$$

where the maximization is written in bold letters because different interpretations of **max** may be used. Accordingly, Step 2 in Algorithm 2 changes to

$$\text{sort } p \in \mathcal{P} : \text{build } S = (p_1, \dots, p_m) : \vec{v}(p_i) \succeq \vec{v}(p_{i+1}) \text{ for all } i = 1, \dots, m$$

where \succeq may be instantiated by different order relations.

There is a large body of literature in applied mathematics, operations research, and decision theory on approaches to solve multi-criteria decision problems (Keeney and Raiffa, 1976; Ehrgott, 2000; Triantaphyllou, 2000). The following two sections present two general approaches, namely hierarchical decision-making and the transformation of multi-attribute into single-attribute decision problems by a combined value function. In both cases, preference information on the individual criteria is incorporated in decision-making with the result that a single solution is returned.

A third direction is known as *multi-objective optimization*, which does not require a trade-off between the criteria, here called objectives, during optimization. Instead, all objectives are simultaneously optimized. As a result, the subset of all of non-dominated solutions is returned. This set is known as the *Pareto optimal set* and sometimes also referred to as the *Pareto-frontier* in the space of all solutions. To obtain a single solution, the user has to pick one solution according to her particular preferences. In the AL scenario, there is, however, no such user in the loop that could make such a decision in each AL iteration. The only user in the loop is the annotator, who has generally no knowledge about the AL procedure and trade-offs between the criteria. Instead, preference information (either static or a dynamically changing over time) must be given to select the first- or $|\mathcal{B}|$ -best examples per iteration, making multi-objective optimization inadequate for the AL scenario.

8.2 Hierarchical Decision-Making

In a straightforward approach to solve MADM problems, the attribute values of the alternatives are compared lexicographically. The order relation \succeq , which we deliberately left unspecified before, is now given by the lexicographic order \geq_{lex} :

$$\vec{v}(p') \geq_{\text{lex}} \vec{v}(p'') :\Leftrightarrow v_i(p') \geq v_i(p'') \text{ for all } i = 1, \dots, q. \quad (8.1)$$

The result is a hierarchical sampling process where the examples are ranked by the attribute-value functions, going through the single attributes in order of descending importance until the ranking is unambiguous.

According to the general AL framework, Step 2 is then

$$p^* = \underset{p \in \mathcal{P}}{\operatorname{arglexmax}} \vec{v}(p)$$

and for batch-mode AL as in Algorithm 2, Step 2 changes to

$$\operatorname{sort} \mathcal{P} : (\mathcal{P}, \geq_{\text{lex}})$$

A prerequisite for this approach is a given ranking among the criteria. Whether such a ranking exists in practise where there are conflicting criteria is, however, questionable. Another problem arises because attribute values of different criteria cannot compensate for each other. Assume criterion a_1 is considered more important than criterion a_2 . In consequence, even an extremely good value $v_{a_2}(p)$ cannot compensate for a slightly worse value $v_{a_1}(p)$.

A special form of hierarchical decision-making (HDM) is the *subset-retaining* variant. As in the general HDM scenario, examples are ranked by subsequent evaluation of the criteria. For each hierarchy level, a subset of the examples is built after ranking according to some threshold. Examples not in this subset are omitted from further evaluation. Retained examples are evaluated anew by the next criterion.

8.3 Multi-Attribute-Value Functions

When trade-offs between all criteria are clearly defined, a common approach to combine the attribute-value functions $v_a(a_d)$ is based on the definition of a *multi-attribute-value* (MAV) function $\phi(\vec{v}(p))$. Using a MAV, the multi-attribute decision problem is eventually reduced to a single-attribute decision problem, optimizing the MAV function value.

As in the HDM approach, the MAV approach also results in a single best solution or a distinct ranking of all alternatives. Depending on the underlying specification of the original MADM problem and the particular MAV function used, not all non-dominated solutions of the Pareto-frontier may be found. This is the case when the Pareto-frontier is non-convex or has non-convex areas (Ehrgott, 2000).

Aggregation of several attributes to one conjoint measure, as done by the MAV functions, is a common technique applied in a wide range of fields. While methodologically often highly similar, different terminology is used for MAV functions, including most prominently “scalarization” in the context of multivariate optimization

(Ehrgott, 2000), “classifier decision combination” in the context of multi classifier systems where predicted labels or confidence values from different classifiers need to be combined into a single outcome (Ho et al., 1994), and “data fusion” or “multiple evidence combination” in the context of information retrieval where results from different query formulations or several ranking criteria need to be combined into a single ordered list (Hsu and Taksa, 2005). In what follows, some MAVs that seem suitable for use in the MCAL scenario are discussed.

Ordinal and Cardinal Value Functions It is often a challenging problem to define a function $v_a(a_d)$. It may be that the value of an attribute can only be measured on an ordinal scale. From an ordinal value function one can only derive an ordered set of alternatives. Most MAVs require cardinal value functions. As a solution, some MAVs originally defined for cardinal value functions can be applied on ranks, too.

Rank-specific Methods Borda count, originally developed in the context of electoral systems, is a popular method used for classifier decision combination. In our MADM scenario, the Borda count $b(v_a(a_p))$ of an example p for criterion a is the number of other examples $p' \in \mathcal{P}, p \neq p'$ ranked below p according to the attribute value of this criterion. The MAV function based on Borda count is given by

$$\phi_{\text{Borda}}(\vec{v}(p)) = \sum_{a=1}^q b(v_a(a_p)). \quad (8.2)$$

Borda count only requires ordinal value functions making it attractive for scenarios where a cardinal value function is hard or impossible to specify.

Maximum over Attribute Values The simple maximum method is defined as

$$\phi_{\text{MAX}}(\vec{v}(p)) = \max_{a=1, \dots, q} (v_a(a_p)) \quad (8.3)$$

This MAV may be reasonable in scenarios where the criteria do not conflict but reinforce each other. While ϕ_{MAX} is defined on cardinal value functions, it can be also used with ordinal value functions translating values into ranks.

Weighted Sum Method The weighted sum MAV is based on a linear combination and given by

$$\phi_{\text{WSM}}(\vec{v}(p)) = \sum_{a=1}^q \gamma_a v_a(a_p) \quad (8.4)$$

where $\sum_{a=1}^q \gamma_a = 1$ and γ_a are the weights of the individual criteria. For ϕ_{WSM} it is important that the single value functions are well-defined. As for ϕ_{MAX} , application on ordinal value functions is possible when values are transferred to ranks. It should be noted that WSM cannot find alternatives that are within non-convex regions of the Pareto-frontier.

Weighted Product Method A MAV based on a non-linear combination is given by the weighted product method

$$\phi_{\text{WPM}}(\vec{v}(p)) = \prod_{a=1}^q v_a(a_p)^{\gamma_a} \quad (8.5)$$

where γ_a are the weights of the single criteria.

8.4 Previous Work on Multi-Criteria AL

This section aims to give a brief overview of the approaches applied. It should be noted, that AL approaches where generally different criteria are fused in a low-level manner into a monolithic criterion – as discussed in Chapter 3 for some approaches to utility functions incorporating both informativeness and representativeness – are not considered to be a case of MCAL. Instead, we consider a problem to be a case of MCAL when separate criteria explicitly need to be considered during selection.

Preservation of representativeness of selected examples, as well as diversity in batch-mode AL, gave rise to a number of MCAL approaches. The incorporation of diversity in the selective sampling process was first proposed by Brinker (2003) for SVMs. In this work, diversity is calculated by the angles between the hyperplane of the previous AL iteration and the hyperplane resulting from the new training data. Combination with the utility score is based on ϕ_{WSM} .

In Kim et al. (2006), an example's diversity is estimated by its maximum similarity to all other unlabeled examples. The entropy-based utility function is applied in an US scenario. Finally, the ϕ_{WSM} is applied to obtain an overall attribute value per example. It is here called maximal margin relevance as the similarity (the negated divergence) is subtracted from the utility score similar to the net-benefit method. The weights γ_a were found empirically and set so that the utility receives a very high impact. Kim et al. (2006) tested their method on biomedical NER; they report slightly improved sampling efficiency with their MCAL approach.

McCallum and Nigam (1998) combined utility (estimated by the Kullback-Leibler divergence to the mean) and representativeness (estimated by density). They applied

ϕ_{WPM} where both criteria obtained equal weights. In the same spirit, representativeness based on the density of an example was also proposed by Tang et al. (2001) in the context of natural language parsing. The density is based on the average distance of an example to all other unlabeled examples. Both criteria are combined by ϕ_{WPM} . Moreover, Zhu et al. (2008c) and Settles and Craven (2008) proposed the same procedure in the context of other application scenarios.

Symons et al. (2006) as well as Shen et al. (2004) consider informativeness, representativeness, and diversity in a MCAL scenario. In Symons et al. (2006), informativeness and representativeness, quantified by similarity to other unlabeled examples, are combined by ϕ_{WPM} . Weights γ_c were again found empirically. Examples sorted by the combined value are reviewed top-down and greedily added to the batch if they respect constraints defined over the diversity of the batch. This method can be considered to be a case of subset-retaining HDM, where the secondary criterion comes with specific constraints. Their evaluation on an NER scenario showed a clear improvement over multi-criteria AL based only on informativeness.

Similarly, Shen et al. (2004) incorporated the same three criteria. Two combination strategies were presented: Firstly, they applied subset-retaining HDM where a subset of examples was selected according to their informativeness score. This subset was clustered and the centroids of the clusters were selected. The cluster centroids were considered most representative and a selection from different clusters leads to a diverse batch. Their second combination strategy is equivalent to the one proposed by Symons et al. (2006). According to their experiments on NER the second strategy performed better.

Few works have studied additional criteria for MCAL, other than representativeness and diversity. For parsing, Becker and Osborne (2005) presented two different utility criteria for AL which are combined using the MAX method. The first criterion is defined as the US-based tree entropy criterion and the second one is based on a parser-error score. For a sequence-selection scenario, Cheng et al. (2008) considered as a secondary criterion the maximal annotation effort that can be spent per AL iteration. This effort per example, which is a sentence as in our NER-specific AL framework, is measured by its length in terms of tokens. Now, MCAL is formulated as a constrained optimization problem with the effort as an inequality constraint and a utility score as the single objective. This can be reformulated as a constraint-based version of HDM. Incorporation of annotation effort in terms of real costs has also inspired approaches to MCAL. This issue is addressed in detail in Chapter 11.

This short literature review provides empirical evidence that our enumeration and classification of approaches to MCAL is valid and useful – the methods applied to MCAL before can be satisfactorily placed in this categorization. To the best of our

knowledge, all published approaches to MCAL are based on one or the other of the two approaches described: HDM or MAVs.

8.5 Summary

This chapter is meant as an introduction to part three of the thesis which addresses the issue of multi-criteria AL. Multi-criteria AL becomes relevant, when other criteria besides the utility of an example should be considered during AL selection. We formally defined the concept of multi-criteria AL as a multi-attribute decision-making problem and described two general methods (HDM vs. MAVs) and specific instantiations of these to solve such problems.

The following three chapters are examples of multi-criteria AL: in Chapter 9, we consider a scenario where the training set should be constituted with respect to several learning problems simultaneously. Chapter 10 covers the issue of how to address and circumvent class imbalance during AL data acquisition time. And finally, in Chapter 11, both utility and costs should be considered during sampling.

Chapter 9

Selective Sampling for Multiple Learning Problems

Default AL selects samples for only one single learning problem. In consequence, when training data for several learning problems is to be created based on AL, each learning problem requires a separate and independent AL-based annotation cycle. Modern HLT systems consist of several components run in a pipelined manner, each solving its own NLP task on the same underlying data. For example, a system populating a biomedical fact database (Buyko et al., 2009) first does some syntactic analysis including, amongst others, statistical parsing, and then turns to the semantics, including NER and relation or event extraction. Given training data is needed for all of the q tasks included and that this material is sampled by standard AL, we would end up with q corpora which cannot be merged as they consist of different examples annotated with respect to different learning problems.

The central question of this chapter is how a combined corpus annotated with respect to several learning problems can be created using AL to select the examples. This is a totally new field of AL and has not been discussed or proposed before. There are two reasons why a combined corpus annotated for various tasks, could be of immediate benefit. Firstly, annotators working on similar annotation tasks – e.g., one task being annotation of named entities and another task the annotation of relations between these entities – might exploit annotation data from one task for the benefit of the other. If for each task a separate corpus is sampled by means of AL, annotators will definitely lack synergy effects and, therefore, annotation will be more laborious and is likely to suffer in terms of quality and accuracy.

Second, in many larger NLP applications classifiers are data-dependent on each other. A classifier might require features as input which are based on the output of preceding classifier. As a consequence, training such a classifier which takes into account several annotation tasks will best be performed on a rich corpus annotated with respect to all input-relevant tasks.

In response to Requirement 6, we here introduce multi-task AL as a new field of AL-related research and propose two straightforward approaches to it. Multi-task AL can be considered as a case of MCAL with all tasks involved being the individual criteria for which examples should be efficiently selected. We evaluate our approaches to multi-task AL in the context of a scenario including NER and syntactic parsing as two rather dissimilar learning problems.

We have recently published most of this work in Reichart et al. (2008).

9.1 Related Work

Multi-task AL has not been addressed as such in literature before. It is, however, based on methods described in Chapter 8. Moreover, multi-task AL is also related to AL sample reusability, discussed in Chapter 7, in the sense that in both scenarios samples are not (directly) drawn with respect to the final consumer.

Recent work on joint inference (McCallum, 2009; Finkel and Manning, 2009) gives evidence that single learning problems support each other. This motivates the need for corpora annotated with respect to a multitude of learning problems.

Finally, multi-task AL should not be confused with multi-task learning.¹

9.2 Problem Definition

We call a scenario where joint training material for multiple learning problems should be provided a *multi-task annotation* scenario. As a primary characteristic, *joint* training material contains annotations for all learning problems of the multi-task annotation scenario. *Multi-task* AL is defined as a scenario where joint training material has to be created for q learning problems $Z_j \in \mathcal{Z}$ based on AL selection.

In contrast to multi-task AL, standard AL selecting samples with respect to a single learning problem Z_j , only. We thus also call it *single-task* AL. In accordance with Chapter 7, in a multi-task annotation scenario, single-task AL constitutes a self-selection scenario for the learning problem focused on during selection. For all other learning problems not considered during selection, single-task AL constitutes a foreign-selection scenario. The goal of multi-task AL is to achieve a sampling

¹Multi-task learning is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It does this by learning problems in parallel while using a shared representation; what is learned for each task can help other tasks to be learned better (Caruana, 1997).

efficiency better than that of random sampling and any foreign-selection for each learning problem in a multi-task annotation scenario.

Multi-task AL can be considered to be a case of MCAL. The criteria to be considered are the q different learning problems which all come together with their respective, single-task AL approaches. Each such AL approach provides utility scores $u_{Z_j}(p)$ for each unlabeled example p under the perspective of the particular learning problem Z_j . The objective of multi-task AL is to select a set \mathcal{B} containing examples that constitute the best compromise of learning problem-specific utility scores $u_{Z_j}(p)$.

For the multi-task AL scenario, we thus have a vector $\vec{v}(p) = (v_{Z_1}(p), \dots, v_{Z_q}(p))$ with attribute-value functions $v_{Z_j}(p)$. Each such function depends on the utility function of the learning problem-specific AL approach. In the simplest scenario, this is $v_{Z_j}(p) = u_{Z_j}(p)$. This simple equation is, however, unlikely to be appropriate because utility scores of specific learning problems are usually not compatible.

Multi-task AL can be difficult when the learning problems under scrutiny are highly dissimilar. As with all MCAL problems, the challenge is to find those examples which on average exhibit high utility values for all learning problems involved. When the learning problems involved tend to produce opposite rankings of the examples according to their utility scores, it is rather unlikely that examples will be found which do serve the requirements for these learning problems well. In such cases, random sampling is still presumably the best option.

9.3 Approaches to Multi-Task Active Learning

In the following sections, two straightforward approaches to multi-task AL are proposed. The motivation for these approaches is to enable experimentation with simple instantiations of multi-task AL and to test whether the overall sampling efficiency in a multi-task annotation scenario can be improved by multi-task AL.

9.3.1 Alternating Task-Specific Selection

The *alternating selection* approach incorporates the utility assessment of several learning problem-specific AL approaches by alternating the task to select for. Thus, each AL iteration constitutes a foreign-selection scenario for $q - 1$ learning problems. According to the importance of a learning problem Z_j , AL selection for this specific problem is performed in a predefined number of consecutive rounds. This enables weighting of the different learning problems by allowing them to guide the AL selection in more or fewer AL iterations.

This simple approach to multi-task AL is a straightforward compromise between the different per-task AL approaches. It requires minimal modification of the general framework for greedy AL (Algorithm 1 on page 32). Basically, Step 2 changes to

$$p^* = \operatorname{argmax}_{p \in \mathcal{P}} \sum_{Z_j \in \mathcal{Z}} \mathbf{1}_{Z_j} v_{Z_j}(p) \quad (9.1)$$

where $\mathbf{1}_{Z_j} = \{0, 1\}$ is an indicator function showing whether selection is done with respect to learning problem Z_j in this AL iteration. Only one of the q indicator functions $\mathbf{1}_{Z_j}$ can be activated per iteration, so that $\sum_{j=1}^q \mathbf{1}_{Z_j} = 1$.

While the alternating selection approach constitutes a single-task AL scenario in each single AL iteration, in the long run it does indeed constitute an approach to multi-task AL. A drawback of this simple approach to multi-task AL is that it does not aim to find examples that are a good compromise for all learning problems involved. In settings where examples that are optimal for one learning problem are not only useless but even adversarial for the learner of another problem, multi-task AL based on alternating selection is thus likely to fail.

9.3.2 Rank Combination

The *rank combination* approach is more directly based on the idea of combining the attribute-value functions $v_{Z_j}(p)$ of all learning problems Z_j . This is done based on a MAV function as proposed in Section 8.3 using the weighted sum method ϕ_{WSM} .

Due to incompatibility, the learning problem-specific utility scores $u_{Z_j}(p)$ are translated into ranks so that the single-attribute-value functions are given by $v_{Z_j}(p) = r(u(p))$. The ranking function $r : \mathbb{R} \rightarrow \mathbb{N}$ assigns higher ranks for higher values of $u(p)$ so that $u(p_i) > u(p_j) \Leftrightarrow R(u(p_i)) > R(u(p_j))$. Step 2 of Algorithm 1 on page 32 now changes to:

$$p^* = \operatorname{argmax}_{p \in \mathcal{P}} \sum_{Z_j \in \mathcal{Z}} \alpha_{Z_j} v_{Z_j}(p) \quad (9.2)$$

where α_{Z_j} specifies the weight of the different learning problems. In the case of tied ranks, we normalize all learning problem-specific ranks to the same range. Depending on the actual utility function applied, tied ranks may occur frequently. This is, for example, the case for $u_{\text{VE}}^{\bar{s}}$ in combination with small committee sizes.

The rank combination approach favors examples which on average are good for all learning algorithms. Examples that are highly useful for one learning problem but rather irrelevant for another problem will not be selected.

corpus	pool set	test set	entity types
WSJ	10,000	1,921	person, location, organization
BROWN	10,000	2,424	

Table 9.1: Overview of simulation corpora used in this chapter. Size of the pool and the test set is given in number of sentences.

9.4 Empirical Assessment

For our experiments on multi-task AL, we considered a two-task scenario that includes NER and syntactic parsing. The tasks are highly dissimilar, thus increasing the potential value of multi-task AL. Moreover, both tasks are subject to intensive research by the NLP community.

9.4.1 Experimental Settings

Corpora For our experiments, we applied two corpora containing both entity and constituent parse annotation.² The WSJ corpus represents the newspaper genre and is based on the Wall Street Journal part of the Penn Treebank (Marcus et al., 1993); the first 10,000 sentences of section 2-21 were used as pool \mathcal{P} , and the 1,921 sentences of section 00 as test set. The BROWN corpus is a mixed-genre corpus and is also based on the respective part of the Penn Treebank. We created a sample consisting of 8 of any 10 consecutive sentences in the corpus. This was done as BROWN contains text from various English text genres, and we did it to create a representative sample of the corpus domains. We finally selected the first 10,000 sentences from this sample as pool set. The 9th of every package 10 consecutive sentences went into the evaluation set which consists of 2,424 sentences. Originally, the Penn Treebank does not contain entity annotations. Thus, we enriched the WSJ and the BROWN corpus with entity annotations (person, location, and organization) as obtained by our CRF-based NE tagger trained on the CoNLL corpus. Table 9.1 summarizes statistics on these corpora.

Active Learning The approaches to multi-task AL described in the previous section are independent of the actual learning algorithm. This is because the approaches, in the manner of MCAL, are a combination of the learning problem-specific utility functions.

²In syntactic analysis, a constituent is a single or a sequence of words which function as an atomic unit in a hierarchical structure such as a syntactic parse tree (Jurafsky and Martin, 2000).

AL for the parsing task is in line with the QbC-based approach described in Reichart and Rappoport (2007). In this work, a committee of 10 parsers was employed and the parser applied is a variant of the Collins parser (Bickel, 2005). The utility function is based on the F-complement u_{FC} (cf. Equation 3.13 on page 39) which calculates the mutual F-score between all committee members. For parsing, the F-score of a sentence is calculated on the recall and precision values of the constituents in the sentence. For consistency, AL for the NER task was also run with QbC-based AL. The u_{VE}^s utility function as described in Chapter 4 was applied since it performed best amongst the utility functions for QbC-based AL.

As in previous experiments, complete sentences were selected. Sentence-level selection granularity is the de-facto standard for AL applied to syntactic parsing. In contrast to the majority of the experiments performed in this thesis, the batch size $|\mathcal{B}|$ was here much higher with 100 examples selected in each AL iteration. Since parser training is extremely complex, AL with small batches would have prohibitively slowed down our experiments. Moreover, AL is started from a random seed set of 200 sentences because parsers need a reasonable amount of information to be able to learn anything useful. In contrast, if the seed set chosen was too small, early AL iterations could exhibit sampling efficiency well below random sampling for the parsing task. Within a corpus we used the same seed set for all experiments. As for alternating selection, the selection leadership changes in each iteration. Similarly, for rank combination equal weights $\alpha_{Z_j} = \frac{1}{|\mathcal{Z}|}$ are assumed for all learning problems Z_j , for simplicity.

9.4.2 Foreign-Selection

The core assumption made in this chapter is that such AL foreign-selection performs poorly in a multi-task AL scenario. This intuition is demonstrated in Figure 9.1 where random and AL foreign-selection sampling efficiency for both final consumers on the WSJ corpus are shown. For both the NER and the PARSE consumer, the respective AL foreign-selection does indeed exhibit poor sampling efficiency equivalent to or even inferior to that of random sampling. Thus, examples that are useful for single-task AL for NER are on average less useful for the PARSE consumer than random examples.

9.4.3 Explicit Selection for Multiple Learning Problems

This section compares the two approaches to multi-task AL, i.e., alternating selection (ALTER-AL) and rank combination (RANKS-AL), with random sampling (RD) and single-task AL for both tasks (NER-AL and PARSE-AL). Figure 9.2 shows the learning

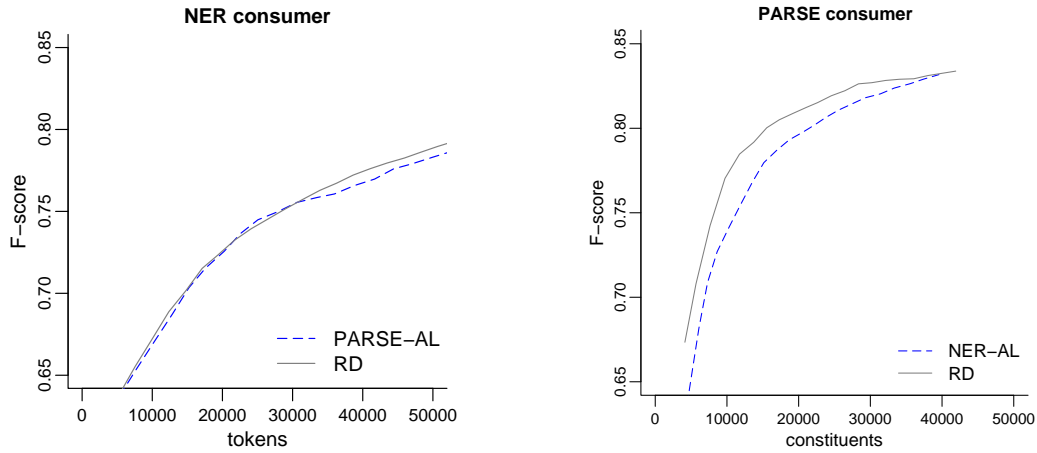


Figure 9.1: Learning curves of foreign- and random selection for NER and PARSE consumer on the WSJ corpus.

curves for these five sampling strategies evaluated both by the NER and the PARSE consumer. For the NER consumer, the F-score is reported as a function on the number of tokens annotated. In contrast, for the PARSE consumer, the number of constituents is reported as a more appropriate cost function.

As expected, self-selection always performs best. As an example, consider the PARSE consumer on the WSJ corpus. Self-selection (PARSE-AL) clearly outperforms random sampling while foreign-selection (NER-AL) is considerably inferior to random sampling so that for the PARSE consumer one should refrain from sampling data with NER-AL.

Both multi-task AL approaches, however, clearly outperform random selection and constitute a real improvement over foreign-selection for both consumers. For the PARSE consumer, both ALTER-AL and RANKS-AL lead to sampling efficiency similar to self-selection. For the NER consumer, although the improvement is less pronounced, the two approaches to multi-task AL still exhibit much better sampling efficiency than foreign- or random selection.

Except for the NER consumer on the BROWN corpus, sampling efficiency of both multi-task AL approaches is very similar. This is a rather unexpected outcome as we assumed that the more sophisticated rank combination approach would outperform the simpler alternating selection protocol. Although there is a slight tendency for RANKS-AL to be better than ALTER-AL, the incorporation of both single-task AL approaches into a combined AL approach appears to be the most important factor.

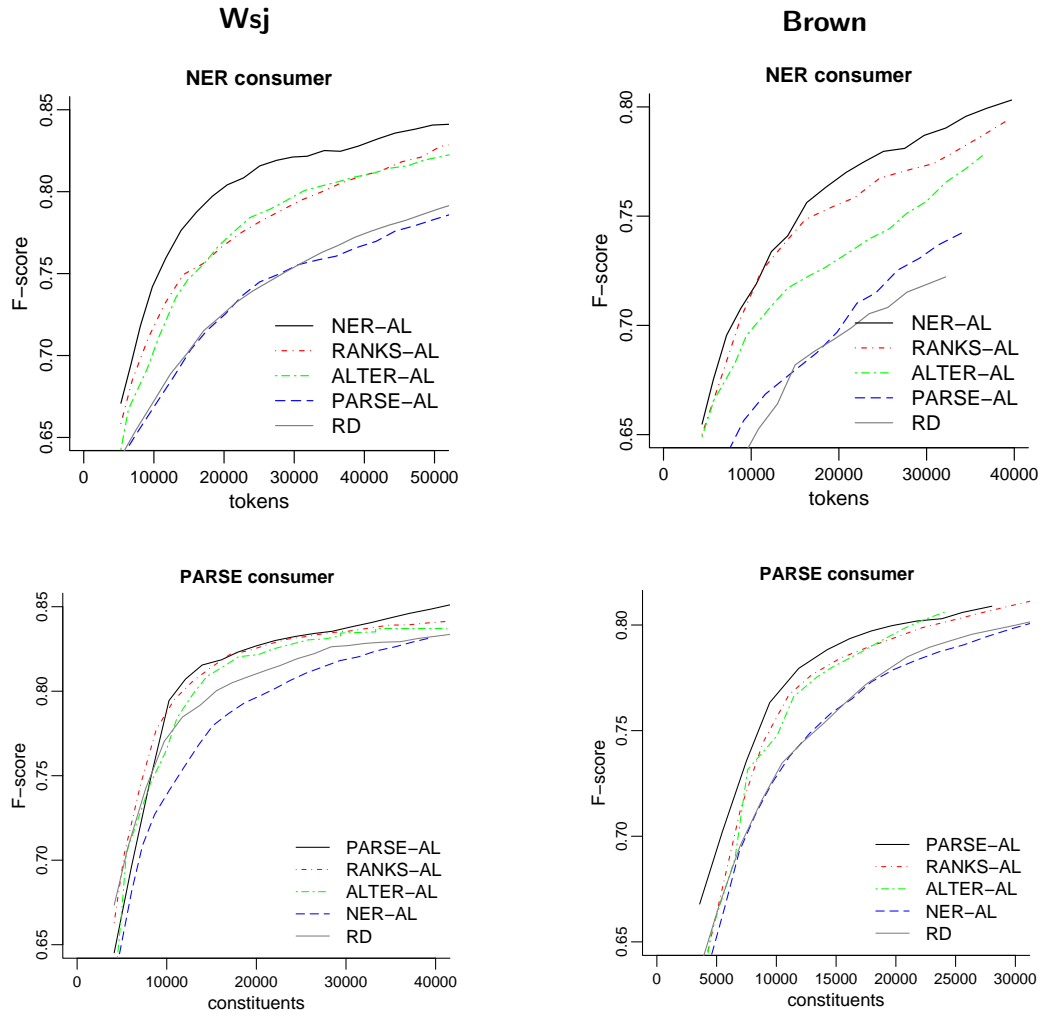


Figure 9.2: Learning curves for the NER and the PARSE consumer.

WSJ corpus				BROWN corpus			
selector	consumer		Σ	selector	consumer		Σ
	NER	PARSE			NER	PARSE	
NER-AL	8.12	-5.36	2.76	NER-AL	8.69	-2.45	6.24
PARSE-AL	0.231	1.94	2.17	PARSE-AL	2.35	4.52	6.87
RANKS-AL	5.79	1.38	7.17	RANKS-AL	9.75	2.99	12.74
ALTER-AL	4.27	-1.23	3.04	ALTER-AL	5.65	1.53	7.18

Table 9.2: RAI scores for NER and PARSE consumer with different selectors and different approaches to multi-task AL. Last column is the overall RAI score, i.e., the unweighted sum over the RAI scores of the NER and the PARSE consumer.

9.4.4 Overall Evaluation

The overall evaluation of a multi-task annotation scenario is delicate and would require a commensurable cost metric for all learning problems involved. Moreover, preference information about the improvement of sampling efficiency for the single learning problems should be given. In our example, a 1 percentage point improvement of the PARSE consumer might count more than the same improvement of the NER consumer – or vice versa, depending on the particular application scenario.

To provide an overall evaluation of multi-task AL in the absence of both true annotation time as a commensurable cost metric and such preference information, the following simplifying assumptions are made: the only commensurable cost metric available for these experiments is the number of sentences selected, which is equal to the AL iterations. Moreover, we assume that improvement by both consumers is considered equally valuable.

Given these assumptions, Table 9.2 shows the overall RAI scores for the AL selection strategies after 10 AL iterations.³ RAI scores indicate poor foreign-selection performance which is consistent to our observations in the previous section. However, RANKS-AL performs better than ALTER-AL now that we evaluate against the number of sentences annotated. Overall, both approaches to multi-task AL still clearly outperform a foreign-selection scenario with RANKS-AL achieving the highest improvements in sampling efficiency. On WSJ, for example, foreign-selection with NER-AL results in a RAI-score of 2.76, which is more than doubled to 7.17 by RANKS-AL.

³RAI scores are calculated in the interval [300; 1300]. This equals the sentences selected in the first 10 AL iterations (starting from a seed set of 200 sentences, 100 sentences selected in each AL iteration). It should be recalled that RAI scores are calculated relative to a random baseline; in consequence, RAI scores for RD are omitted.

9.5 Summary and Conclusions

This chapter has described and defined the problem of multi-task AL. In the multi-task AL paradigm, examples are actively selected with respect to multiple learning problems instead of a single one, as it is usually done in the context of AL. To the best of our knowledge, multi-task AL has so far not been recognized or described as a relevant problem in literature.⁴ We proposed two approaches to multi-task AL, i.e., alternating selection and rank combination, and tested these approaches in a two-task scenario that includes NER and syntactic parsing.

In a multi-task annotation scenario, foreign-selection is likely to result in poor overall sampling efficiency. While highly useful for the learner of the problem involved in AL selection, the same examples may on average be less useful than randomly drawn ones. This chapter has gathered empirical evidence for this assumption in our two-task scenario. Moreover, our experiments provide promising evidence that the simultaneous consideration of all relevant learners during AL selection can be a much better choice than foreign-selection in multi-task annotation scenario. This proved true even for alternating selection, a rather simple approach to multi-task AL.

Future investigations will have to focus on the question of whether the positive results observed in our orthogonal (i.e., highly dissimilar) two-task scenario will also hold for a more realistic (and maybe more complex) multi-task scenario where learning problems are more similar and more than two learning problems are involved.

Our attempt to provide an overall evaluation showed that the quantification of sampling efficiency in a multi-task AL scenario is not straightforward and requires additional scenario-specific knowledge. Firstly, either exchange rates between the annotation cost metrics or, even better, commensurable costs, are needed. Secondly, information is required on the application-specific benefits of performance improvement per learning problem. Exchange rates on costs and preference information usually are inherently tied to the specific task, domain, and application.

Future work in this context would require studies on how to quantify the efforts needed for the annotation of a textual unit of choice (e.g., tokens, sentences, constituents) with respect to different annotation. The next chapter, although it is not in the context of a multi-task annotation scenario but rather standard, single-task AL, discusses further issues of cost-sensitive evaluation and realization of AL.

⁴Our joint work on multi-task AL with Roi Reichart and Ari Rappoport from the Hebrew University is published in Reichart et al. (2008).

Chapter 10

Reducing Class Imbalance during Active Learning Sampling

In many NLP tasks, the classes to be dealt with often are heavily imbalanced in the underlying data set. This holds for example, for NER, where imbalance between the different entity classes occurs especially when semantically general classes (such as person names) are split into more fine-grained and specific ones (actors, politicians, sportsmen, etc.). Classifiers trained on such skewed data tend to exhibit poor performance for low-frequency classes. Since rare information carries the potential to be particularly useful and interesting, performance should to a certain extent be tuned more in favor of minority classes, at the risk of penalizing the overall outcome.

Class imbalance and the resulting effects on learning classifiers from skewed data have been intensively studied in recent years. Common ways to cope with skewed data include different re-sampling strategies and cost-sensitive learning (Japkowicz, 2000; Chawla et al., 2002; Elkan, 2001). It has been shown that AL itself can also be used to leverage class imbalance: The class imbalance ratio of data points close to the decision boundaries is typically lower than the imbalance ratio in the complete data set (Ertekin et al., 2007) so that AL automatically creates data with more balanced classes.

The focus of this chapter is whether this natural characteristic of AL can be intensified to obtain even more balanced data sets (Requirement 7). Thus, instead of first acquiring possibly skewed data in a possibly expensive acquisition process and then, once the data is completely annotated and available, applying typical approaches to overcome class imbalance, we consider the problem of reducing class imbalance right upfront during the AL-driven data acquisition process. The ultimate goal of this research is to find approaches to compiling more balanced data sets upfront during annotation time when AL is used as a strategy to acquire training material.

The general idea is to guide AL, so as to focus more on the minority class. In consequence, examples for the minority class with good learning utility might be

preferred over examples for the majority with extremely high learning utility. Considering class imbalance during AL sampling can be realized as a multi-criteria AL (MCAL) approach with the main criterion being learning utility of an example and the second criterion the overall class imbalance ratio among the labeled data.

This chapter proposes and compares four approaches to reducing the class imbalance upfront during AL-driven data acquisition. The first two approaches are not realized by MCAL. However, the third and fourth are formulated as a MCAL problem. While the third approach is based on a form of hierarchical decision making, the fourth one is completely different from the MCAL approaches described in Chapter 8 because it incorporates the second criterion in a low-level manner into the first criterion.

Most of this work has been previously published in Tomanek and Hahn (2009a).

10.1 Related Work

There is a vast body of literature on the class imbalance problem in the ML community. Common ways for coping with skewed data include re-sampling strategies such as under- and over-sampling or generative approaches (Japkowicz, 2000; Chawla et al., 2002), and cost-sensitive learning (Elkan, 2001). AL has already shown to be capable of reducing class imbalance because it selects data points near the decision boundaries where data points are more balanced (Ertekin et al., 2007). As a result, AL provides the learner with a sub-sample of the available data. In this scenario, AL does not have a human annotator in the loop as the data is already labeled.

Class imbalance is typically addressed in scenarios where (large) numbers of fully labeled examples are readily available. Our scenario is different in that we start from unlabeled data and use AL to select the examples to be labeled by a human annotator. In our scenario, class imbalance should be avoided upfront during the process of selecting and annotating training data.

There is little work on the combination of AL and remedies to class imbalance at annotation time. Zhu and Hovy (2007) combined AL and re-sampling strategies, including under- and over-sampling, for word sense disambiguation. In each AL iteration, examples were selected on the basis of a default AL scheme. Accordingly, either examples of the majority class were discarded, or examples of the minority class were replicated. While the authors reported that under-sampling was inefficient because some examples were directly discarded after they had been manually labeled beforehand, positive evidence was found for over-sampling combined with AL. While Zhu and Hovy only considered re-sampling techniques, we study different approaches

to address class imbalance during the AL selection process and formulate it as a problem of MCAL.

Recently, Bloodgood and Shanker (2009b) proposed a method to address class imbalance by cost-weighted SVMs during the AL process. Their method is very similar to our fourth approach where the utility function is modified as to additionally incorporate information on the class imbalance. Bloodgood and Shanker derived class-specific cost factors from the class imbalance ratio observed on a small random data sample. A limitation of their approach is that it relies on a cost-sensitive learning algorithm. In contrast, our approach is independent of the respective learning algorithm and can thus be applied more flexibly.

10.2 Approaches

For simplicity, our approaches to reducing class imbalance during AL were developed and tested in the context of a two-class scenario. The less frequent class is called *minority class*, the other one *majority class*. Transferred to the NER scenario, we consider a scenario with two *entity* classes (the *minority entity class* and the *majority entity class*), and an OUTSIDE class. Our interest lies in the distribution of the entity classes in terms of class imbalance between these two classes, however, not in the class imbalance including the OUTSIDE class. The OUTSIDE class is, of course, still subject to annotation and classifier learning. The *entity class ratio* is defined as the number of majority class entity mentions divided by the number of minority class entity mentions.

In the NER-specific AL scenario, a sentence-level selection granularity is applied. This selection granularity constitutes a more complex scenario because a sentence consists of multiple units which require classification. Thus, unlike most class imbalance studies where “pure” minority or majority class examples were considered, sentences simultaneously contain instances of the minority *and* the majority class.

A first assumption as to how to address class imbalance during AL was that the AL process might be positively affected by a seed set optimized so as to contain a balanced ratio of minority and majority class instances. The basic idea of such optimized seed sets is that the AL process would be permanently guided towards the less frequent class by its early “announcement”. Such an announcement is not given in the case of a randomly sampled seed set and because the seed set is likely not to contain a single instance of the minority class.

As reported in Chapter 4 and (Tomanek et al., 2009), in another context we tested how much AL is affected by different seed sets in the NER scenario. Due to the

co-selection effect, unfavourable seed sets show little influence on the performance of AL in the NER scenario with sentence-level selection granularity. Analogously, in the NER-specific AL framework, balanced seed sets performed very well only in very early iterations but then fell back to the performance of AL with randomly drawn seed sets.

As the application of optimized seed sets alone is not a sustained method to reduce class imbalance, four approaches are described which directly influence the sampling process with the aim of focusing more on examples containing the minority class.

10.2.1 Modification of the Selector

To focus on specific classes during AL, the selector can be trained on labeled data for these classes only. In consequence, the learning utility of an example is evaluated only with respect to these classes. However, after selection, examples are still annotated with respect to all classes. In our two-class scenario this means that only the minority class is considered during selection itself, but the selected examples can also be assigned a majority class label by the annotator.

We call this approach *minority class-focused AL*. While one would expect it to produce good results for the minority class, high penalties are likely for the majority class. In the NER scenario with sequence-selection, however, sentences containing minority class entity mentions in many cases also contain majority class mentions due to the co-selection effect. As a result, training material acquired by minority class-focused AL may contain information on the majority class as well. Whether such information is sufficient to learn a consumer with good performance on the majority class is evaluated in the next section.

10.2.2 Post-Processing of the Selection

Re-sampling strategies, including over- and under-sampling, are common practise to tackling the class imbalance problem when passively learning from skewed data sets (Provost, 2000). Both methods can also be applied in a straightforward manner as a post-processing step to AL selection. After the manual annotation step in each AL iteration, either examples for the minority class are over-sampled (e.g., by simple replication), or examples of the majority class are discarded to achieve a more balanced entity class ratio among the current selection.

Under-sampling appears to be disadvantageous when AL is applied in order to acquire labeled training data: After having expended human labeling effort on the

selected examples, some of these would be immediately discarded in the post-processing step. Over-sampling, in contrast, is more favourable. It does not render previous annotation effort superfluous and comes with no extra costs. For word sense disambiguation, Zhu and Hovy (2007) showed that AL combined with over-sampling as a post-processing step applied in each AL iteration can considerably increase performance.

Due to the sampling granularity in the NER scenario, we do the over-sampling on the sentence level. All sentences selected within one AL iteration which contain at least one instance of the minority class are duplicated. This approach is called *over-sampling during AL*. We deliberately refrained from the specification of a fixed over-sampling ratio for the NER scenario. When sentences are considered as atomic units and can only be replicated as a whole, such a ratio between the minority and the majority class is almost impossible to realize.

Additional instances of the minority class are bought at the price of increasing the number of majority class instances. Experiments show, however, that the class ratio is shifted in favor of the minority class when sentences containing information on the minority class are duplicated.

10.2.3 Hierarchical Selection

The NER-specific AL approach by default selects a set of $|\mathcal{B}|$ sentences with the highest utility scores. In the presence of a highly skewed class distribution, this set presumably contains only few instances of the minority class, if any. This fact directly results in class imbalance in the training set and may also have a misleading effect on further AL iterations reinforcing the tendency to favor sentences with instances of the majority class.

As a means of addressing this problem with MCAL, we propose the hierarchical consideration of the individual criteria during selection. Here, the first and most important criterion is the learning utility of an example, and as a second criterion, class distribution is considered. The objective of an AL iteration is thus changed to selecting a batch set of $|\mathcal{B}|$ examples which have a high estimated training utility accompanied by a possibly balanced class distribution of the batch set. This is achieved by altering Step 3 of the NER-specific AL framework (Algorithm 2 on page 55).

From the set of unlabeled examples \mathcal{P} , which is sorted in descending order based on the learning utility $u(p)$, a candidate set \mathcal{S} of the top-ranging $|\mathcal{B}| \cdot s$ examples from \mathcal{P} is selected. In a greedy fashion, $|\mathcal{B}|$ examples are moved from \mathcal{S} to \mathcal{B} so that the entity class ratio in \mathcal{B} is maintained close to 1.

We call this approach *balanced-batch AL*. Apparently, a completely balanced batch cannot be attained in every AL iteration. The maximum achievable level of balance depends on the size of the candidate set \mathcal{S} and on the number of minority and majority instances contained therein. An overly large candidate set leads to devaluation of the learning utility as first criterion. However, when chosen too small, the second criterion hardly comes into play.

10.2.4 Combined Metric

A disadvantage of balanced-batch AL is that it does not take into account the learning utility scores of the examples within the candidate set \mathcal{S} : Any example within this set which optimizes the class ratio may be selected, irrespective of its actual utility score. To explicitly consider learning utility, we propose to combine the two criteria learning utility and the effect on class distribution into a single score and select according to it. This can be categorized as an approach to MCAL based on MAV functions.

However, in contrast to the generic MAV functions described in Chapter 8, we interweave its utility and effect on class distribution on a low level dependent on the actual utility function chosen. The Vote Entropy-based utility function u_{VE} is well-suited for combination with a criterion of class distribution. It should be remembered, that the VE is described for QbC-based AL (Equation 3.12 on page 39). It estimates the utility as the disagreement of the committee. Therefore, the distribution of the classes predicted by the single committee members is considered.

The new MAV function $\phi(u_{VE'}(p, \mathcal{C}, \vec{b}))$ is given by a modification of the VE so that it incorporates a vector with class-specific boosting factors $\vec{b} = (b_1, \dots, b_{|\mathcal{Y}|})$, with $b_i \geq 1$:

$$u_{VE'}(p, \mathcal{C}, \vec{b}) = - \sum_{i=1}^{|\mathcal{Y}|} b_i \cdot \frac{V(y_i, x)}{|\mathcal{C}|} \log \frac{V(y_i, x)}{|\mathcal{C}|} \quad (10.1)$$

Note, that $u_{VE'}$ is formulated independently of the NER-specific scenario. For application for sentence-level selection granularity, $u_{VE'}$ scores of all tokens of a sentences are aggregated by the mean-average as discussed in Chapter 4.

The votes on a specific class y_i are given more importance by setting the class boosting factor $b_i > 1$. A value of $b_i = 1$ does not affect the disagreement, while a higher value of b for the minority class accounts for our intuition that an example where at least one committee member assumed the minority class should be considered more useful and thus result in a higher combined score. Moreover, the less certain the

committee is about a particular class label (i.e., fewer committee members voting for this class), the more this boosting factor affects the overall score of the example. If all committee members agree in their prediction, the overall combined score is 0, irrespective of the chosen boosting factor.

As default boosting factor b_{\min} for the minority class in the NER scenario, we choose the entity class ratio divided by the average number of tokens per minority class entity mention. The normalization by entity length is reasonable, since the boosting factor applies to the token level, while the entity class ratio is calculated with reference to the entity mention level.

10.3 Experiments

This section reports on the empirical evaluation of the AL approaches to reducing class imbalance. Experiments are performed in the context of the NER task where imbalance between the different entity classes especially occurs for fine-grained categorizations. The four approaches, *viz.*, minority class-focused AL (AL-MINOR), balanced-batch AL (AL-BAB), AL with boosted disagreement with the default boosting factor (AL-BOOD), and over-sampling during AL (AL-OVER), are compared to random sampling (RD) and the unmodified AL approach using the $u_{VE}^{\bar{s}}$ utility function (AL-DEF).

10.3.1 Experimental Settings

Corpora For these experiments, we chose corpora different from the standard corpora applied throughout this thesis. The reason is that those corpora do not exhibit a large enough class imbalance. For the NER scenario, we found a high imbalance ratio especially in the biomedical domain where entity classes are often more fine-grained and class imbalance between entities occurs in a more pronounced way than in the newspaper material.

We focus on scenarios with *two* entity classes only, namely one majority and one minority entity class. Our first data set (MAL) is based on the annotations of the PENNBIOIE corpus (Kulick et al., 2004). From the rich set of entity classes considered in the original PENNBIOIE corpus, we kept only two malignancy entity classes for our experiments. The majority class is based on PENNBIOIE’s malignancy-type annotations, the minority class combines all classes describing malignancy stages.

Our second data set (TF) is based on the GENEREG corpus which is annotated with genes involved in the regulation of gene expression (Hahn et al., 2008). Here,

	MAL	TF
sentences	11,164	4,629
tokens	277,053	139,600
OUTSIDE tokens	257,173	136,266
majority class tokens	18,962	3,152
minority class tokens	918	182
majority class entities	9,321	2,776
minority class entities	604	179
entity class ratio	15.43	15.51

Table 10.1: Characteristics of the simulation corpora.

all entity mentions except those labeled as *transcription factor* (majority class) and *transcription cofactor* (minority class) were removed. Table 10.1 summarizes the characteristics of both data sets. While the MAL corpus is much larger than TF, both sets have approximately the same entity class imbalance ratio of 15.5.

Active Learning Recall that in the context of the AL scenario, our interest lies in the distribution of the entity classes in terms of class imbalance between these two classes, but not on the class imbalance including the OUTSIDE class. The OUTSIDE class is of course annotated, learned, and predicted by the selectors and consumers. However, the *entity class ratio* is defined as the number of majority class entity mentions divided by the number of minority class entity mentions.

Committee-based AL using the $u_{VE}^{\bar{s}}$ utility function is applied except for AL-BOOD. In each AL iteration $n = 25$ sentences are selected and AL is started with a randomly drawn seed set of 25 sentences. Reported results are averages over 30 independent runs. For each run, we randomly split the data set into a pool from which AL selects and a gold standard for evaluation. On MAL, 90% of the sentences are used as the AL pool and 10% for evaluation. Due to the smaller size of the TF data set, 30% of the sentences were used for evaluation to obtain a reasonable coverage of minority class entity mentions in the gold standard. The findings from Chapter 7 on the reusability of samples obtained from AL with a MaxEnt-based selector by a CRF-based consumer are exploited here to speed up the experiments.

For AL-BAB, $s = 5$ so that the size of the candidate set \mathcal{S} is set to $25 \cdot 5 = 125$. Experimental validation showed this to be a good trade-off between both criteria. For AL-BOOD, both the OUTSIDE class and the majority entity class are not boosted so that $b_{maj} = b_{outside} = 1$ and for the minority entity class the default boosting factor is used.

Macro vs. Micro F-Score In previous chapters, sampling efficiency of AL approaches in the NER scenario was evaluated in terms of the micro F-score. The micro F-score is an average where each single class F-score is weighted proportionally to the number of examples for this class in the gold standard. Thus, the micro F-score is dominated by the classification performance on the high-frequency classes and implicitly assumes these classes to be of higher importance. The macro F-score, in contrast, shows how well a classifier performs across all classes, as it is calculated as the unweighted average over the single class F-scores (cf. Section 2.3).

The following experiments are evaluated with the macro F-score because we assume the minority and the majority class to be equally important.

10.3.2 Results

10.3.2.1 Re-balancing during Data Acquisition

To determine whether, and if so, to what extent the approaches presented are suitable for guiding the AL process towards the minority class in our NER scenario, we analyzed the effect of the protocols with respect to the entity class ratio, the number of minority and majority class entities in the labeled data, and performance in terms of different F-scores. While the primary goal is to increase the macro F-score, we also show the F-scores for both classes separately as a more detailed picture of the effects of the alternative protocols.

Figure 10.1 shows the entity class ratio yielded by each protocol at different token positions. The ratio for random sampling roughly corresponds to the data sets' overall entity class ratio of 15.5. As already shown before in a different context (Ertekin et al., 2007), AL-DEF also shifts the ratio in favor of the minority class to values of about 11 on both data sets. While AL-BOOD achieves a very low ratio in early AL iterations, the ratio increases in later iteration rounds. Only AL-MINOR and AL-OVER maintain a low ratio over many AL rounds.

Figure 10.1 also depicts the absolute numbers of entity mentions of the minority and majority class. Using AL-MINOR or AL-BOOD on the MAL corpus, most entity mentions of the minority class have been found and annotated after 30,000 tokens. As only few sentences containing minority class entity mentions remain in the unlabeled remainder of the corpus, only a minor performance increase on this class can be expected from that point onwards. At the time when AL-BOOD cannot find many more sentences with minority class entities, the number of majority class entity mentions selected increases. The same pattern, although much more pronounced due to the smaller size of the corpus, holds for the TF data set where for AL-BOOD most

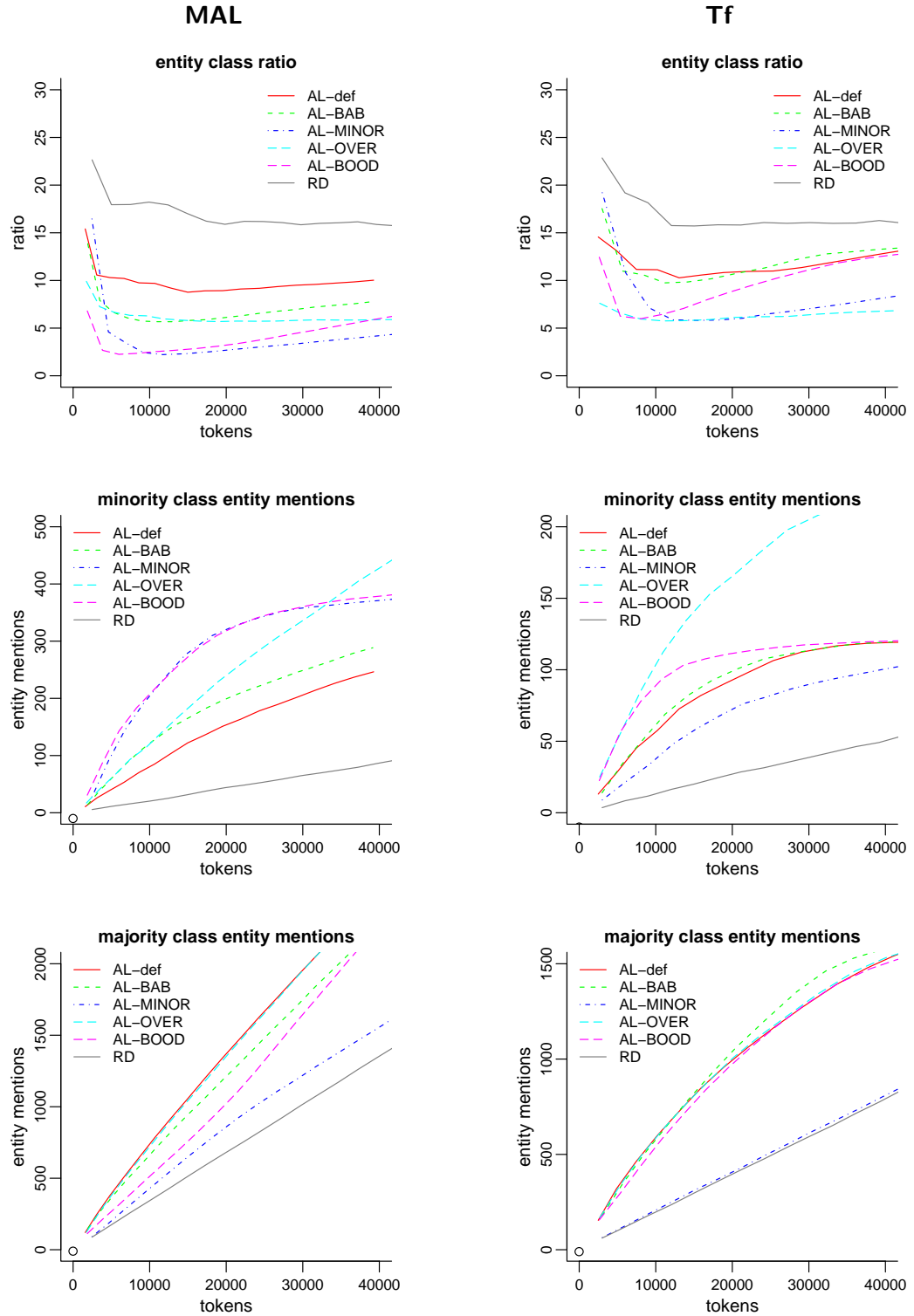


Figure 10.1: Entity mention statistics.

MAL corpus				TF corpus			
AL strategy	F-scores			AL strategy	F-scores		
	min.	maj.	macro		min.	maj.	macro
AL-DEF	44.80	6.27	18.13	AL-DEF	182.50	22.46	56.14
AL-BAB	55.58	5.69	21.05	AL-BAB	179.40	21.40	54.67
AL-MINOR	72.33	-1.25	21.60	AL-MINOR	59.33	1.15	14.31
AL-OVER	45.48	4.89	17.39	AL-OVER	209.30	21.76	60.71
AL-BOOD	72.32	1.11	23.15	AL-BOOD	229.50	20.71	63.42

Table 10.2: RAI scores for the different AL strategies. Scores are calculated for the three F-scores: minority class, majority class, and macro F-score.

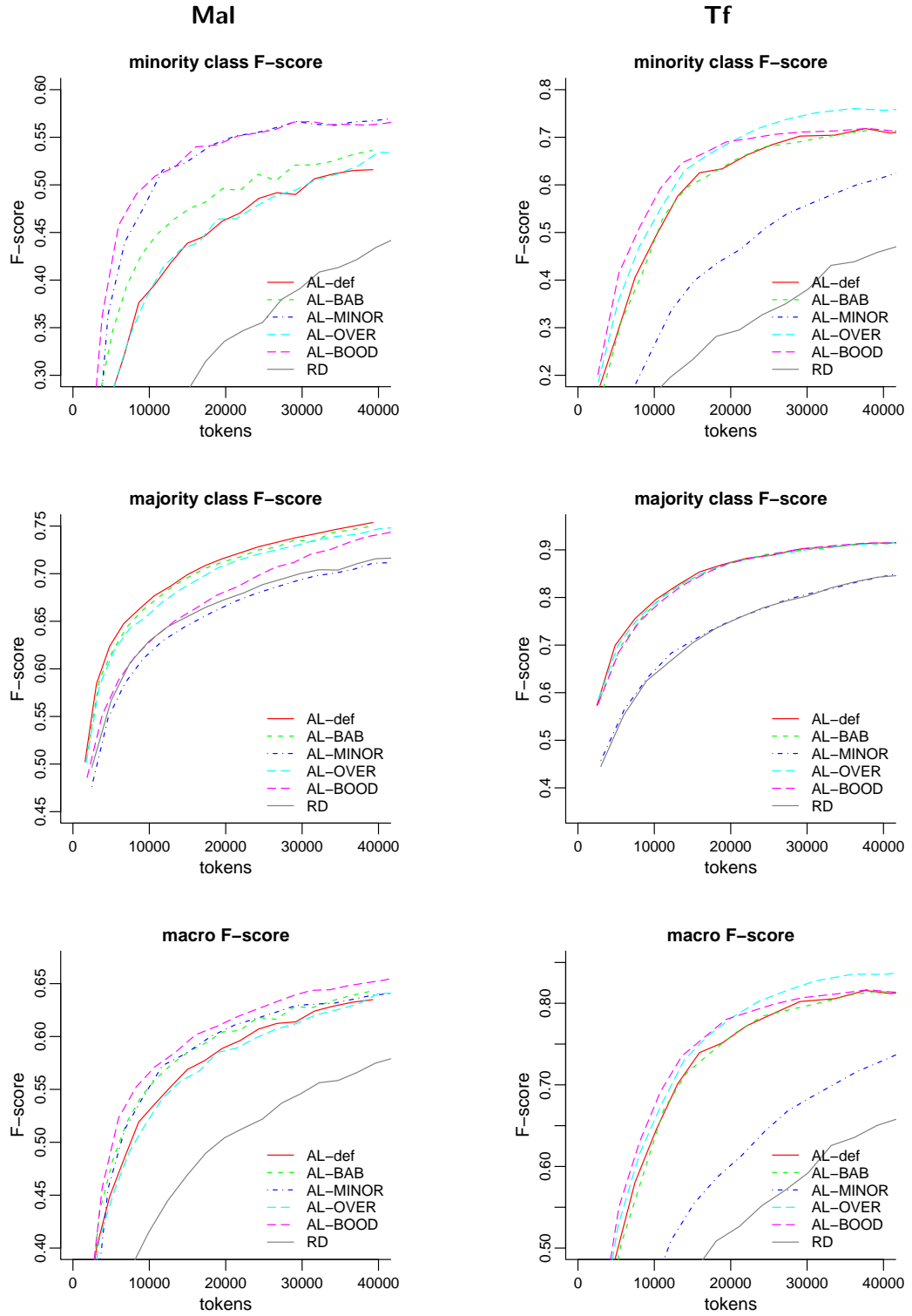
entity mentions from the minority class have been found after 15,000 tokens.¹ On MAL, AL-OVER exceeds the number of minority class entities selected by AL-BOOD after 30,000 tokens as for AL-BOOD the pool is then exhausted with respect to the minority class entity mentions. On TF, this happens in very early AL iterations. While AL-OVER considerably increases the number of minority class entity mentions on both data sets, it does not affect the number of majority class entity mentions. AL-MINOR results in high numbers on minority class entities on MAL, comparable with those of AL-BOOD; on TF, however, this protocol performs even worse than AL-DEF. Overall, we see that the different protocols have an effect on the number of minority class entity mentions and, as a result, on the entity class ratio.

Figure 10.2 shows learning curves for the minority class, the majority class, and the macro F-score and Table 10.2 shows the respective RAI scores. RAI scores on MAL were calculated in the interval of [10,000; 30,000] tokens, while for TF we chose the interval of [5,000; 20,000] tokens. We deliberately limited the range covered by RAI scores to that part of the AL process where not all minority class entity mentions were not yet found.

On the MAL data set, AL-OVER does not perform considerably different from AL-DEF in terms of minority class or macro F-score but causes slight deterioration of the majority class F-score. AL-BAB slightly increases the minority and macro F-score, with minor losses on the majority class F-score. While both AL-BOOD and AL-MINOR result in a steep increase of the minority class F-score, AL-BOOD performs better than AL-MINOR in terms of the macro F-score. This is because AL-MINOR harms the majority class F-score considerably so that it is almost as low as RD.

On the TF corpus, until about 15,000 tokens AL-BOOD outperforms AL-OVER in terms of minority class (and also slightly macro) F-score. After that point AL-OVER

¹After splitting TF into the AL pool and the gold set, only about 100-110 entity mentions of the minority class remain in the pool.



170 Figure 10.2: Learning curves of minority class, majority class, and macro F-score.

takes over. This can be explained by the phenomenon we have already observed in Figure 10.1: Almost all of the minority class entity mentions of the AL pool have been found and selected by AL-BOOD. Obviously, AL-OVER can and does outperform the other protocols as it is less restricted by the small overall number of minority class entity mentions. AL-BAB does not have any relevant effect on the TF data set, its RAI score is even slightly below that of AL-DEF. AL-MINOR performs very poorly here – worse than AL-DEF, even on the minority class F-score. We observed that a randomly compiled seed set is quite disadvantageous here as it hardly contains any minority class entity mentions. Thus, in early AL iterations, the binary classifiers employed in the committee for AL-MINOR mostly predict the OUTSIDE class and thus hardly disagree, so that AL-MINOR rather resembles a random selection mechanism in early AL iterations until – by chance – some more sentences with minority class entity mentions are selected. Only AL-MINOR has a relevant (negative) effect on the majority class; all other protocols did not affect the majority class F-score.

In all our experiments, we applied AL-BOOD with the default boosting factor for the minority class determined by the heuristic described in Section 10.2.4 so that $b_{\min} = 10.15$ for MAL, and $b_{\min} = 15.25$ for TF. Performance of AL-BOOD with different values for b_{\min} is discussed below in Section 10.3.2.4.

The RAI scores in Table 10.2 once again underline that the gains in sampling efficiency for the minority class are much higher than the losses for the majority class. Depending on the actual trade-off between model performance on the minority and majority class, the overall assessment of the evaluated approaches might change.

10.3.2.2 Re-balancing after Data Acquisition

Instead of addressing class imbalance *during* data acquisition, one could also apply AL-DEF to select the data to be annotated and once the data is available apply re-sampling techniques to address class imbalance. To study whether in our scenario re-sampling *within* the AL loop is more appropriate than re-sampling applied as a *post-processing* step, additional experiments were performed.

We ran the selection by AL-DEF, then over-sampled during evaluation time in the same manner as we did in AL-OVER, that means all sentences containing the minority class were duplicated at each evaluation position. A direct comparison of over-sampling during AL selection (AL-OVER) and delayed over-sampling after AL selection (AL+OVER) reveals that on both data sets AL-OVER performs worse than AL + OVER (see RAI scores in Table 10.3). We conclude, that AL-OVER does not

MAL corpus				TF corpus			
AL strategy	F-scores			AL strategy	F-scores		
	min.	maj.	macro		min.	maj.	macro
AL-OVER	45.48	4.891	17.39	AL-OVER	209.3	21.76	60.71
AL+OVER	52.72	5.988	20.44	AL+OVER	211.6	22.65	62.12

Table 10.3: RAI scores for over-sampling *during* (AL-OVER) and *after* (AL+OVER) the AL process. Scores are calculated for the three F-scores: minority class, majority class, and macro F-score.

enforce a special “guidance” effect on the AL selection process and thus – if over-sampling is applied at all – it is better to do so once the labeled data is available and not during annotation time.

10.3.2.3 Re-balancing during and after Data Acquisition

Our first experiments showed that AL-BOOD outperforms AL-OVER on the MAL corpus (cf. Table 10.2). However, to profit from both AL-BOOD’s ability to select sentences containing many minority class entities and from the fact that over-sampling can help out when the overall number of minority class entities in a corpus is extremely limited or even exhausted, we combine both protocols (AL-BOOD + OVER).

RAI scores comparing AL-BOOD, AL+OVER, and the combination AL-BOOD+OVER are given in Table 10.4. Figure 10.3 shows respective learning curves of the macro F-score. On both corpora, AL-BOOD+OVER outperforms “pure” AL+OVER as well as “pure” AL-BOOD. The beneficial effects of the combination on the MAL corpus come into play only in later AL iterations (20,000 to 40,000 tokens), whereas on the TF corpus, the combination improves the performance particularly on early to medium AL iterations (up to 20,000 tokens). In later AL rounds the performance of AL + OVER is not exceeded.

With AL-BOOD+OVER, the macro F-score improved on both corpora compared to AL-DEF. On MAL, the macro F-score after 40,000 tokens was increased from 63.74 (AL-DEF) to 66.3 (AL-BOOD+OVER) and in order to reach AL-DEF’s macro F-score of 63.74, we need only approximately 25,000 tokens using AL-BOOD+OVER, which comes to a saving of over 40%. Similarly, on TF we obtain a macro F-score of 83.7 instead of 81.4 and also save about 40% annotation effort to yield AL-DEF’s performance.

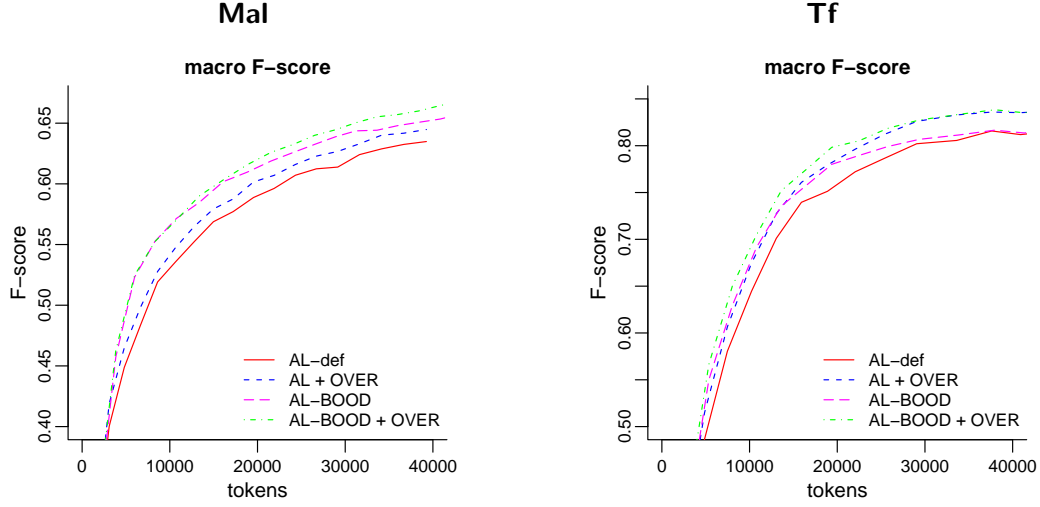


Figure 10.3: Learning curves for the combination of AL-BOOD and over-sampling.

10.3.2.4 Different Boosting Factors

Previously, we applied AL-BOOD with the default boosting factor for the minority class determined by the heuristic described in Section 10.2.4. To study the effect of a particular value for b_{\min} on AL-BOOD, further experiments with different values for b_{\min} were run. At the same time, b_{outside} and b_{maj} were kept at a value of 1. Table 10.5 shows RAI scores for very low ($b_{\min} = 2$) to extremely high ($b_{\min} = 50$) boosting factors. On both data sets, the boosting factor determined by the heuristic did not result in the best performance. Instead, a lower factor of $b_{\min} = 8$ would have performed slightly better in terms of macro F-score. Furthermore, the actual choice of b_{\min} (as long as a minimum of $b_{\min} \geq 4$ is given) is not very relevant in terms of macro F-score. Considering the minority class F-score, however, another factor value than the one obtained by the simple heuristic really pays off.

10.3.3 Discussion

While improvements on the minority class F-score clearly come at the cost of diminished performance on the majority class, our experiments showed overall gains in terms of an improved macro F-score using AL-BOOD in a two-entity-class setting. AL-BOOD only works reasonably well when a sufficient number of examples is available for the minority class. As for TF, AL-BOOD improves upon the minority class F-score in early iterations, only. In real-world annotation settings, however, large

MAL corpus			
AL strategy	F-scores		
	min.	maj.	macro
AL-OVER	52.72	5.988	20.44
AL-BOOD	72.32	1.11	23.15
AL-BOOD+OVER	76.21	0.837	24.23

TF corpus			
AL strategy	F-scores		
	min.	maj.	macro
AL-OVER	211.60	22.65	62.12
AL-BOOD	229.50	20.71	63.42
AL-BOOD+OVER	248.70	20.74	67.33

Table 10.4: RAI scores for the combination of AL-BOOD and over-sampling after AL (AL-BOOD+OVER). Scores are calculated for the three F-scores: minority class, majority class, and macro F-score.

numbers of unlabeled examples are typically available so that it is quite unlikely that the AL pool could be exhausted with respect to one entity class. While both AL-MINOR and AL-OVER only performed well on one data set, AL-BOOD was always amongst the best-performing protocols.

Our experiments indicate that our heuristic to determine the boosting factor for the minority class was a good start. However, more sophisticated ways to determine such a factor are necessary to yield optimal results. A good boosting factor depends on several influencing factors, including corpus-specific characteristics (average sentence length, number of entities per sentence, number of tokens per entity, difficulty of learning each class, whether sentences contain, on average, exclusively one entity class, etc.), as well as application-specific considerations (misclassification costs and acceptable trade-off between gains on the minority class and losses on the majority class). In real-world annotation projects, the value of the boosting factor might also change over time. If a severe class imbalance is ascertained after several AL rounds, one might adjust the boosting factors accordingly.

10.4 Summary and Conclusions

In this chapter, several approaches to reducing class imbalance upfront during AL-driven data acquisition were proposed and compared. Our experiments revealed that

MAL corpus				TF corpus			
b_{\min}	F-scores			b_{\min}	F-scores		
	min.	maj.	macro		min.	maj.	macro
1 (=AL-DEF)	44.80	6.27	18.13	2 (=AL-DEF)	182.50	22.46	56.14
2	54.91	5.80	20.95	2	211.30	22.47	61.56
4	66.67	4.83	23.91	4	227.90	21.13	63.28
8	71.70	2.92	24.16	8	237.10	20.65	64.57
10.15*	72.32	1.11	23.15	15.25*	229.50	20.71	63.42
16	74.28	1.25	23.90	16	231.70	20.51	63.46
50	72.89	0.48	22.96	50	228.20	20.41	62.77

Table 10.5: RAI scores for AL-BOOD with different boosting factor values for the minority class. AL-BOOD with $b_{\min} = 1$ accords to AL-DEF. Boosting factors marked with asterisk (*) are obtained by the heuristic. Best macro F-score is highlighted.

class imbalance can indeed effectively be reduced, accompanied by an increase in the performance of classifiers with respect to minority class and the preservation of good overall performance in terms of the macro F-score. One of our approaches based on the low-level incorporation of a boosting factor into the $u_{VE}^{\bar{s}}$ utility function (AL-BOOD), combined with over-sampling after the data acquisition process turned out to be the best of several alternatives tested. While AL-BOOD has been formulated in the context of the $u_{VE}^{\bar{s}}$ utility function, it could also be applied to a non-committee based approach as well as with other utility functions.

Given the Zipfian nature of natural language, class imbalance is a ubiquitous problem for NLP and by no means limited to task of NER in the biomedical application domain which we have deliberately chosen as our experimental framework. Future work might study the application of AL-BOOD to other NLP tasks such as relation extraction, a task usually subject to immense and adversarial class imbalance.²

²Relation extraction is usually formulated as a binary classification problem. The negative class is usually highly overrepresented.

Chapter 11

Cost-Sensitive Active Learning

Although it has been shown that AL can yield impressive reductions in annotation effort, AL has not yet become an accepted annotation strategy. One reason might be because annotation practitioners are in doubt about the true efficiency of AL. In the previous chapters of this thesis, as well as in most works on AL, annotation effort is estimated by simple cost measures such as the number of tokens being annotated.

For annotation practitioners, however, it is not the sheer corpus size (in terms of number of sentences or other units) but instead the time actually needed to annotate such a corpus that is the ultimate question of interest (Requirement 8). This chapter studies AL in the context of real annotation effort. Section 11.2 describes a resource containing both NE annotations and information on the annotation time required, which was constructed to enable such studies. In Section 11.3, the approaches to AL presented in Chapter 4 and 6 are evaluated against true annotation time on this new resource. Finally, Section 11.4 reports on novel approaches to making AL cost-sensitive.

11.1 Previous Work

Cost-sensitive AL (CSAL) is a relatively new field of research into AL. Several approaches to and consideration on CSAL have been published recently, many of them in the context the of the Workshop on Cost-Sensitive Learning¹ and the Workshop on Active Learning for Natural Language Processing.² It should be noted that in this thesis, CSAL focuses on data acquisition costs that result from human labeling effort. In contrast, some works have also focused on misclassification costs and making AL sensitive to them (Margineantu, 2005; Sheng and Ling, 2007).

¹Held in conjunction with the Neural Information Processing Systems Conference 2008.

²Held in conjunction with the 2009 Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies.

An early work on appropriate cost metrics for AL was published by Becker and Osborne (2005). The authors examined whether AL, while decreasing the sampling size, possibly increased annotation effort. In context of a real-world AL annotation project, it is demonstrated that the actual sampling efficiency measure for an AL approach depends on the cost metric being applied. In a companion paper, Hachey et al. (2005) studied how sentences selected by AL affected the annotators' performances both in terms of time needed and annotation accuracy, and came to the conclusion that selectively sampled examples are, on average, more difficult to annotate than randomly sampled ones.

This observation questions the widespread assumption that all examples exhibit the same annotation effort. In a study of annotation costs in four real-world text and image annotation tasks, Settles et al. (2008) collected empirical evidence for high variability of annotation costs. Such insights constitute the motivation for approaches to CSAL.

Only a few studies have been published on CSAL. Haertel et al. (2008b) and Settles et al. (2008) proposed to make a standard AL approach cost-sensitive by normalizing the utility score by annotation time. Donmez and Carbonell (2008) proposed an approach to CSAL that they formulated as a constraint-based optimization problem: instead of sampling a fixed number of examples in each iteration, a budget based on available annotation cost was defined as a constraint. The goal was then to maximize the utility of the selected examples under the budget constraint. In the same spirit as the two previously mentioned works, Donmez and Carbonell also divided the utility by the monetary costs.

Similarly, our procedure to aggregate utility scores, which were calculated on the token-level into an overall utility score for a sequence (Equation 4.1 on page 57), also constitutes a normalization of the utility by cost. In this scenario, cost is, however, assumed to be the number of tokens per sentence in order to simplify matters. Normalization of utility by cost has already been applied by Melville et al. (2005) in the context of AL-based feature-value acquisition.³ Moreover, one of our proposed approaches to CSAL is also based on this idea but generalizes it to make it more applicable to arbitrary utility functions (Section 11.4 below).

Approaches to CSAL based on the net-benefit, which is defined as cost subtracted from utility, were proposed by Vijayanarasimhan and Grauman (2009) for object recognition in images and by Kapoor et al. (2007) for voice message classification. The benefit, or utility, of an example is estimated by its potential to reduce the model's generalization error, i.e., the reduction of risk given that this example was

³AL-based feature-acquisition aims at improving a model's performance by filling missing values of the features in the labeled training data.

available in the training set. Risk is assumed to be expressed in monetary units (for example by a monetary misclassification cost matrix) so that the difference between risk reduction and labeling cost is essentially the net-benefit per example.

Liu et al. (2008) presented an approach to AL for land-cover classification where labeling an example involved physically traveling to a location to determine the ground truth in terms of soil conditions. In contrast to the previously mentioned papers, Liu et al. assume annotation cost to be dependent on previous labeling actions. In their specific scenario of land-cover classification, such dependencies arose from traveling activities. CSAL with dependent labeling costs was tackled by combining US-based AL with approaches to solving the traveling salesman problem.

Few attempts have been made recently to build estimators for annotation costs. Ringger et al. (2008) trained a simple linear regression model based on sentence length and the expected number of false predictions for part-of-speech annotation. A more sophisticated model based on a larger number of features for different text and image annotation tasks was presented by Settles et al. (2008). Most recently, Arora et al. (2009) proposed a cost model for movie review annotation. The moderate performance of the three cost models demonstrates that the features used can only explain a small portion of the annotation costs.

In the absence of highly accurate cost models, simulation of CSAL requires information about the true annotation cost. As the involvement of human annotators in all CSAL experiments is far too costly, corpora extended with annotation time information are an essential resource. Up until now, most experiments with CSAL have either been run with simplistic cost models (Kapoor et al., 2007), synthetic costs (Donmez and Carbonell, 2008), or on home-grown corpora of limited size with coarse-grained time annotation measurements (Settles et al., 2008). To date, no generally approved corpora with time information are available. The corpus proposed in the next section, addresses this shortcoming and extends the existing MUC7 corpus with respect to annotation time measurements.

11.2 The MUC7_T Corpus

This section reports on the re-annotation of selected types of NEs from the MUC7 corpus. The focus of this annotation endeavour is on recording the time needed for the linguistic process of NE annotation. Annotation times taken on two basic annotation units – sentences *vs.* complex noun phrases – are measured. The resulting corpus, MUC7_T, couples common NE annotation meta-data with a time stamp

reflecting the time measured for the underlying decision making.⁴ As a reference baseline for cost-sensitive annotation strategies, as well as for learning accurate cost models, our major requirements for such a time annotated corpus are its size and coherence. The annotation level for which cost information is available is also crucial – document- or sentence-level data might be too coarse for some applications.

To meet the requirement for size and coherence, we decided on the MUC7 base annotations. Time stamps are added to two levels of annotation granularity: sentences and complex noun phrases. The resulting corpus is a valuable resource enabling empirically grounded studies of selective sampling techniques in the context of linguistic annotation processes, such as the experiments performed in this chapter.

More details on the MUC7_T corpus are described in Tomanek and Hahn (2009c) and Tomanek and Hahn (2010).

11.2.1 Annotation Procedure

Our annotation initiative constitutes an extension of the NE annotations (ENAMEX) of the MUC7 corpus (cf. Section 4.3.2) covering three types of NEs, *viz.* persons, locations, and organizations. We instructed two human annotators, both advanced students of linguistics with good English language skills, to re-annotate the MUC7 corpus for the ENAMEX subtask. To be as consistent as possible with the existing MUC7 annotations, the annotators had to follow the original guidelines of the MUC7 NER task. For ease of re-annotation, we intentionally ignored temporal and number expressions (TIMEX and NUMEX).

MUC7 covers three distinct document sets for the NER task. We used one of these sets to train the annotators and to develop the annotation design, and another one for our actual annotation initiative, which consists of 100 articles reporting on airplane crashes. We split lengthy documents (27 out of 100) into halves to make them fit the annotation screen without the need for scrolling. Furthermore, we excluded two documents due to excessive length as they would have required overly many splits. Our final corpus contains 3,113 sentences (76,900 tokens).

Time-stamped ENAMEX annotation of this corpus constitutes MUC7_T, our extension of MUC7. Annotation time measurements were made on two syntactically different *annotation units*: (a) complete sentences and (b) complex noun phrases. The annotation task was defined in such a way as to assign an entity type label to each token of an annotation unit. The use of *complex noun phrases* (CNPs) as an alternative

⁴Such time stamps should not be confounded with the annotation of temporal expressions in MUC7 or more advanced meta-data using TIMEML, as used for the creation of the TimeBank (Pustejovsky et al., 2003).

annotation unit is motivated by the fact that in MUC7 the syntactic encoding of NE mentions basically occurs through nominal phrases. CNPs were derived from the sentences' constituency structure obtained from a syntactic parser (Ratnaparkhi, 1999) to determine top-level noun phrases. To avoid overly long phrases, CNPs dominating special syntactic structures, such as co-ordinations, appositions, or relative clauses, were split up at discriminative functional elements (e.g., a relative pronoun) and these dominated elements were eliminated. An evaluation of the CNP extractor on ENAMEX annotations in MUC7 showed that 98.95% of all entities were completely covered by automatically identified CNPs. For the remaining 1.05%, parsing errors were the most common source of incomplete coverage.⁵

While the annotation task itself was “officially” declared to yield only annotations of NE mentions within the different annotation units, we were primarily interested in the time needed for these annotations. For precise time measurements, so-called *annotation examples* were shown to the annotators one at a time. An annotation example consists of the chosen MUC7 document with one annotation unit (sentence or CNP) selected and highlighted. Only the highlighted part of the document could be annotated and the annotators were asked to read only as much of the context surrounding the annotation unit as was necessary to make a proper annotation decision. To present the annotation examples to annotators and allow for annotation without extra time overhead for the “mechanical” assignment of entity types, the annotation GUI is controlled by keyboard shortcuts. This minimizes annotation time compared to mouse-controlled annotation, such that the measured time reflects only the amount of time needed for taking an annotation decision.

In order to avoid learning effects on the part of annotators on originally consecutive syntactic subunits, we randomly shuffled all annotation examples so that subsequent annotation examples were not drawn from the same document. Hence, annotation times were not biased by the order of appearance of the annotation examples. Annotators were given blocks of either 500 CNP- or 100 sentence-level annotation examples. They were asked to annotate each block in a single run under noise-free conditions, without breaks and disruptions. They were also instructed not to annotate for excessively long stretches of time in order to avoid effects of tiredness making time measurements unreliable.

We compared the annotation results of annotator A and B on 5 blocks of sentence-level annotation examples created during training. Annotation performance was measured in terms of *a*) Cohen's kappa coefficient κ on the token level and *b*) the F-score against MUC7 annotations. The annotators A and B achieved $\kappa_A = 0.95$

⁵The CNP extractor was developed as part of the a master thesis which was supervised by the author of this thesis (Lichtenwald, 2009). For a more detailed description and evaluation of the CNP extractor, please refer to this master thesis and Tomanek and Hahn (2009c).

number of sentences	3,113
number of tokens (in all sentences)	76,900
number of CNPs	15,203
number of tokens (in all CNPs)	45,097
number of entity mentions in all sentences	3,971
number of entity mentions in all CNPs	3,937
number of sentences with entity mentions	63%
number of CNPs with entity mentions	23%

Table 11.1: Characteristics of the MUC7_T corpus.

and $\kappa_B = 0.96$, and $F_A = 0.92$ and $F_B = 0.94$, respectively. They exhibit an inter-annotator agreement of $\kappa_{A,B} = 0.94$ and an averaged mutual F-score of $F_{A,B} = 0.90$. These numbers reveal that the task was well-defined and that the annotators had internalized the annotation guidelines sufficiently well to produce valid results. Moreover, an analysis of the annotation performance over time showed it to be stationary – no general trend in annotation performance over time could be observed.

11.2.2 Corpus Statistics

Table 11.1 summarizes statistics on the time-stamped MUC7 corpus. About 60% of all tokens are covered by CNPs (45,097 out of 76,900 tokens), showing that sentences are to a large extent made up from CNPs. Still, removing the non-CNP tokens markedly reduces the number of tokens to be considered for entity annotation. CNPs cover slightly less entities (3,937) than complete sentences (3,971), a marginal loss only. On average, sentences have a length of 24.7 tokens, while CNPs are rather short with 3.0. However, CNPs vary tremendously in length, with the shortest ones having only one token and the longest ones (mostly due to parsing errors) spanning over 30 (and more) tokens. Extremely long CNPs are mostly due to parsing errors.

Figure 11.1 depicts the length distribution of sentences and CNPs showing that a reasonable proportion of CNPs have less than five tokens, while the distribution of sentence lengths almost follows a normal distribution (at least for lengths between 1 and 50 tokens). While 63% of all sentences contain at least one entity mention, only 23% of CNPs contain entity mentions. These statistics show that CNPs are generally rather short and a large fraction of CNPs does not contain entity mentions at all. We may hypothesize that this observation will be reflected by annotation times.

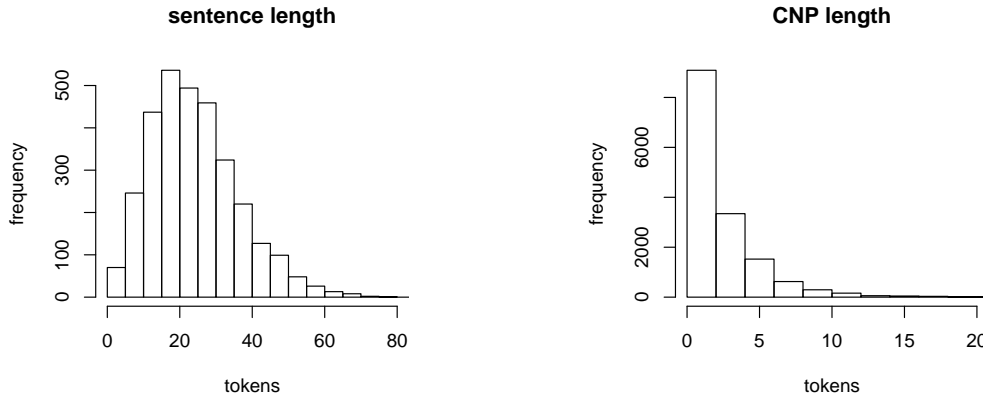


Figure 11.1: Length distribution of sentences and CNPs.

11.2.3 Annotation Time Analysis

Figure 11.2 shows the average annotation time per block of CNPs and sentences. Considering the CNP-level annotations, there is a learning effect for annotator B during the first 9 blocks. After that, both annotators are approximately on a par regarding annotation time. For sentence-level annotations, both annotators again yield similar annotation times per block, without any learning effects. Similar to the annotation performance, the analysis of annotation time shows that the annotation behavior is largely stationary (excluding the first rounds of CNP-level annotation) which allows single time measurements to be interpreted independently of previous time measurements. Both time and performance plots show that there are blocks that were generally more difficult or easier than other ones, because both annotators operated in tandem.

As we have shown, inter-annotator variation of annotation performance is moderate. Intra-block performance, in contrast, is subject to high variance. Figure 11.3 shows the distribution over both annotators' annotation times by boxplots. As for the sentence-level annotations, the median of annotation time is 4.5 seconds (4.1s) for annotator A (B), the shortest time is 0.8s (0.1s), and the longest time is 46s (51s). As for CNPs, time investment is even more skewed with a median of 0.94s (0.91s), a minimum time of 0.06s (0.03s), and a maximum of 118s (97s). To summarize, both for the sentence- and the CNP-level annotations, a large proportion of the respective units can be done with a low time investment but there are also numerous instances where annotation becomes extremely costly. These numbers provide ample evidence for the assumption that the time needed to annotate a particular unit varies greatly (regardless of the individual annotators involved) and that the naïve assumption of uniform costs is untenable. This has already been observed by Settles et al. (2008).

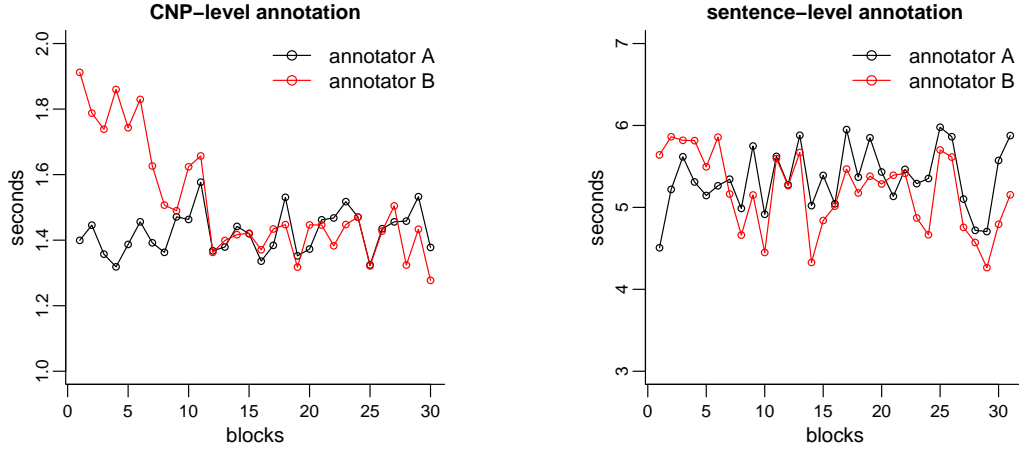


Figure 11.2: Average annotation times per block.

An initial manual analysis revealed that CNPs with very low annotation times are mostly short and consist of stop words and pronouns only, or are otherwise simple noun phrases with a surface structure incompatible with entity mentions (e.g., all tokens are lower-cased). Here, humans can quickly exclude the occurrence of entity mentions which results in low annotation times. CNPs that took an excessively long time (more than 6s) were outliers, indicating distraction or loss of concentration. Times between 3s and 5s were basically caused by semantically complex CNPs.

11.3 Evaluation of Active Learning with Real Costs

With $\text{MUC7}_{\mathcal{T}}$, we can now evaluate sampling efficiency of AL with real annotation costs. We do this for default, i.e., fully-supervised AL (FuSAL) and for semi-supervised AL (SeSAL).

11.3.1 Default Active Learning

We re-ran the experiments of Chapter 4 with the best-performing utility functions $u_{\text{VE}}^{\bar{s}}$, u_{LC}^s , and $u_{\text{MA}}^{\bar{s}}$ on the $\text{MUC7}_{\mathcal{T}}$ corpus. As a new cost measure alongside the token count measure (cf. Equation 4.11 on page 59), the time needed for the annotation was considered. For the experiments, the “true” annotation costs as stored in $\text{MUC7}_{\mathcal{T}}$ were taken. The experimental settings were the same as in Chapter 4, reported results are an average over 20 independent runs, 20 sentences were selected

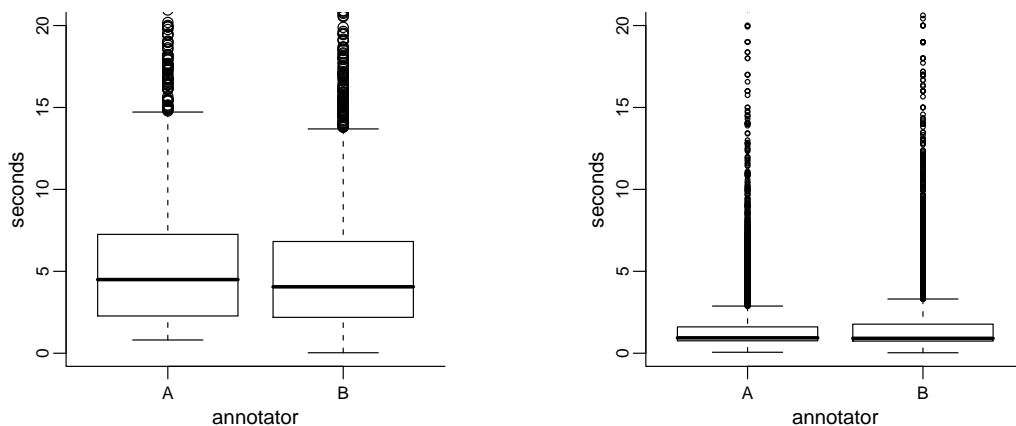


Figure 11.3: Distribution of annotation times over sentences (left) and CNPs (right).

in each AL iteration, and AL was started from a random seed of 20 sentences. Results on $\text{MUC7}_{\mathcal{T}}$ cannot, however, be directly compared to our previous experiments on the MUC7 corpus due to a lower corpus size (76,900 vs. 78,329 tokens) and a lower number of entity classes (3 vs. 7) in $\text{MUC7}_{\mathcal{T}}$.

For this experiment we took the time stamps on the sentence level from the $\text{MUC7}_{\mathcal{T}}$ corpus. While the $\text{MUC7}_{\mathcal{T}}$ corpus has time measurements for two annotators, for the rest of this chapter we report the results of time measurements only for annotator A. Scores for annotator B were very similar and showed exactly the same trends. Figure 11.4 shows the learning curves for the AL experiments, with both the token and the real annotation time cost measures. Additionally, Table 11.2 reports on the resulting reduction of annotation effort for the three utility functions for a target performance of $F^* = 0.89$.

AL still outperforms random sampling when the cost measure is the actual time needed for annotation. The relative cost reduction for annotation time is generally lower than for the token cost metric: While $u_{\text{MA}}^{\bar{s}}$ saves 61.7% of the tokens to be annotated to yield an F-score of 0.89, this translates into a saving of only 46.9% of annotation time. This confirms the findings of Hachey et al. (2005) that selectively sampled examples are on average harder to annotate than randomly sampled ones.

However, even given the real annotation time, AL clearly pays off. This finding is in contrast to the findings of Settles et al. (2008) where US-based AL for relation extraction and sentence classification did not perform better than random sampling when evaluated against real costs. Our positive findings might be due to the NER scenario, which already in previous chapters turned out to be very suitable for AL.

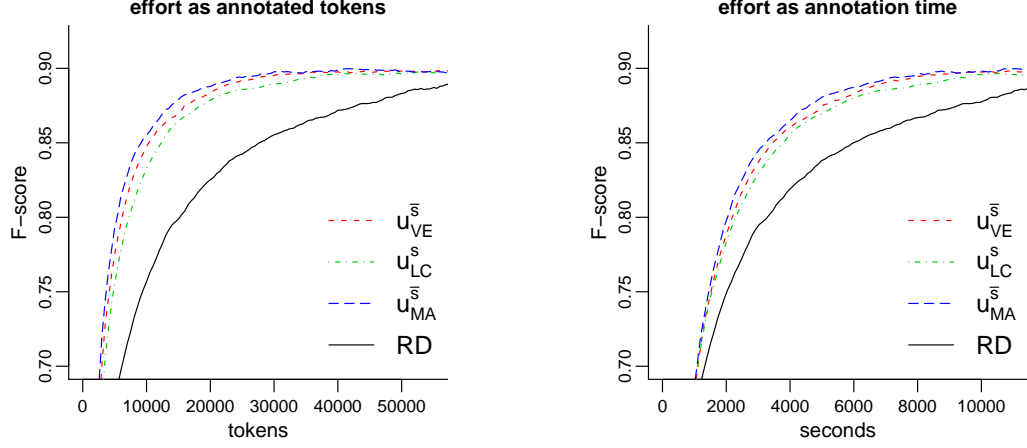


Figure 11.4: Learning curves for evaluation of default AL evaluation against the number of tokens and the true annotation time on $\text{MUC7}_{\mathcal{T}}$.

AL strategy	tokens	seconds
$\Delta\text{CFP}_{F^*}(u_{\text{VE}}^{\bar{s}}, \text{RD})$	58.2 %	44.2 %
$\Delta\text{CFP}_{F^*}(u_{\text{LC}}^s, \text{RD})$	45.8 %	32.9 %
$\Delta\text{CFP}_{F^*}(u_{\text{MA}}^{\bar{s}}, \text{RD})$	61.7 %	46.9 %

Table 11.2: Percentage reduction of annotation effort over random sampling RD for a target performance of $F^* = 0.89$.

As an additional finding it should be noted that also for annotation time $u_{\text{MA}}^{\bar{s}}$ performs best, closely followed by $u_{\text{VE}}^{\bar{s}}$. Both utility functions outperform u_{LC}^s , a sequence confidence-based utility function. As shown in Chapter 4, such utility functions select longer sentences. While longer sentences take longer to annotate overall, this does not pay off in terms of higher gains in classifier performance.

11.3.2 Semi-Supervised AL

One motivation to record annotation time on the CNP level in $\text{MUC7}_{\mathcal{T}}$ was to allow for an annotation time-based evaluation of our approach to SeSAL (cf. Chapter 6). For this evaluation, SeSAL experiments were re-run on the $\text{MUC7}_{\mathcal{T}}$ corpus.⁶

⁶As described in Chapter 6, SeSAL was run with a threshold $t = 0.99$ and a delay rate $d = 0$; the u_{LC}^s utility function is applied both for semi- and fully-supervised AL.

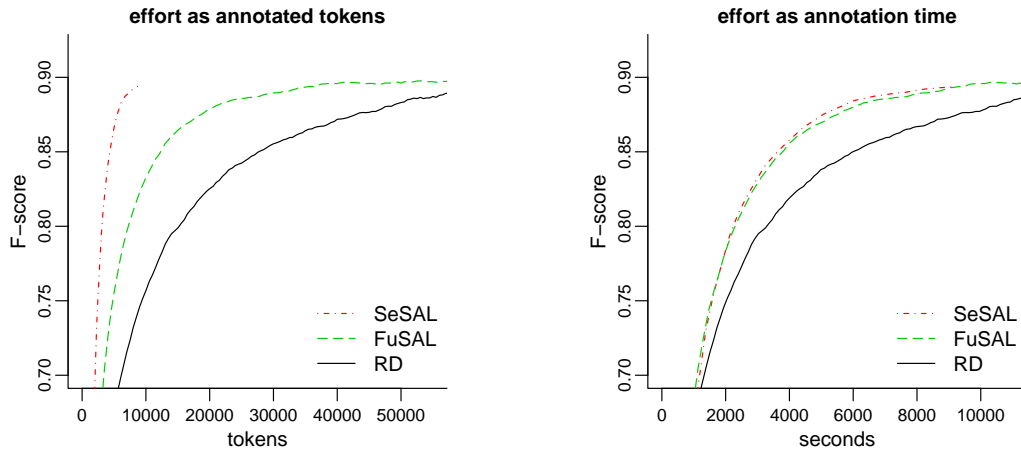


Figure 11.5: Learning curves for SeSAL and FuSAL with number of tokens annotated and real annotation time.

Whenever a CNP contains at least one token classified as uncertain and to be manually annotated by SeSAL, this CNP’s annotation time is considered when calculating the overall time of the respective sentence. The summation of all the annotation times of CNPs that have been gone through by annotators is based on the pessimistic assumption of no synergy effects being obtained by annotating multiple CNPs per sentence: the time required to annotate a CNP within a sentence is considered independent of other CNPs already annotated in the same sentence. To avoid too pessimistic estimates, the annotation time recorded per sentence in $MUC7_{\mathcal{T}}$ is taken as the maximum time even if the cumulated CNP-level times would exceed this. However, the annotation time that we account for here per sentence still constitutes an extremely conservative upper bound.

Figure 11.5 shows the evaluation of SeSAL both in terms of number of tokens and real annotation time measured as described above. In terms of tokens SeSAL saves about 75 % on $MUC7_{\mathcal{T}}$ compared to its fully-supervised counterpart. However, in terms of annotation time, SeSAL only marginally reduces the annotation effort compared to FuSAL: $CFP_{F^*}(\text{SeSAL}) = 8,641$ vs. $CFP_{F^*}(\text{FuSAL}) = 8,728$ with $F^* = 0.89$

These rather poor results need to be put into the perspective of the extremely pessimistic estimation of annotation time. Furthermore, this also shows that the evaluation of AL *simulations* with real annotation times constitute a challenging endeavour as it may not be clear how to measure annotation time – especially, when selected examples are only partially annotated and no human annotators should be involved in the simulations.

11.3.3 Conclusions

Overall, this section shows that the results of AL also depend greatly on the metric used to account for the annotation costs. This has also been addressed by Becker and Osborne (2005) and Haertel et al. (2008a). Evaluated against annotation time, SeSAL did not perform better than its fully-supervised counterpart which, however, yields high reductions in terms of real annotation time. The next section is devoted to the question of how AL can be made intrinsically cost-sensitive.

11.4 Cost-Sensitive Active Learning

CSAL is a typical scenario of multi-criteria AL where two contradictory criteria – utility and cost, here – have to be considered at the same time. In the following, three simple approaches to CSAL are proposed and evaluated on the MUC7_T corpus.

11.4.1 Approaches to Combining Utility and Cost

The following three approaches are based on adaptations of the MCAL methods described in Chapter 8 to the two contradictory criteria utility and cost.

Cost-Constrained Sampling CSAL can be realized in its most straightforward manner by simply constraining the sampling to a particular maximum cost c_{\max} per example. Examples are first ranked by their primary criterion, the utility, and in a second step the $|\mathcal{B}|$ top-ranked examples from this ranked list with costs below c_{\max} are selected. Cost-constrained sampling (CCS) is a simple form of subset-retaining HDM (cf. Section 8.2). In a more efficient manner, CCS can be implemented as a pre-processing step where all examples $p \in \mathcal{P}$ for which $\text{cost}(p) > c_{\max}$ are removed from \mathcal{P} . The unmodified NER-specific AL framework can then be applied.

A shortcoming of CSS is that it does not allow any form of compensation between utility and cost. Thus, an exceptionally useful example with a cost slightly above c_{\max} will be rejected. Another critical issue is how to set c_{\max} . If set too low, the pre-filtering of \mathcal{P} constitutes a strong restriction of selection options when only few examples remain in $|\mathcal{P}|$. If set too high, the cost-constraint is ineffective.

Linear Rank Combination A simple form of the weighed sum method ϕ_{WSM} , known as *net-benefit* and often applied in decision theory, combines criteria of benefit and cost. Given the attribute-value functions $v_{\text{benefit}}(p) = u(p, \theta)$ and $v_{\text{cost}}(p) = -1 \cdot \text{cost}(p)$, the net-benefit is defined as

$$\begin{aligned}\phi_{\text{NB}}(p) &= v_{\text{benefit}}(p) + v_{\text{cost}}(p) \\ &= u(p, \theta) - \text{cost}(p).\end{aligned}$$

Net-benefit has been applied to CSAL in a scenario where both benefits and costs were given as a monetary unit and could thus be directly compared (Kapoor et al., 2007). If the same unit of measurement is not used, a transformation function between benefit and cost must be found, which can be a difficult.

In our scenario, benefits measured by utility scores and costs measured in seconds are incommensurable. For this scenario, it is currently unclear how to express utility in monetary costs or vice-versa. Instead, we transform utility and cost information into ranks to which ϕ_{WSM} can be applied.

The attribute-value function for the learning utility is thus given as $v_{\text{utility}}(p) = r(u(p, \theta))$. The ranking function $r : \mathbb{R} \rightarrow \mathbb{N}$ assigns higher ranks for higher values of $u(p, \theta)$ so that $u(p_i, \theta) > u(p_j, \theta) \Leftrightarrow R(u(p_i, \theta)) > R(u(p_j, \theta))$. For costs, $v_{\text{cost}}(p) = R'(\text{cost}(p))$ with R' as the reversed ranking function so that higher cost values are assigned lower rank numbers. The linear rank combination (LRK) is defined as

$$\phi_{\text{LRK}}(\vec{v}(p)) = \alpha \cdot v_{\text{utility}}(p) + (1 - \alpha) \cdot v_{\text{cost}}(p)$$

where α is a weighting term. In the CSAL scenario, where utility is the primary criterion, $\alpha > 0.5$ seems reasonable. Moreover, as costs and utility are contradictory, allowing equal impact for both criteria with $\alpha = 0.5$, it may be difficult to find appropriate examples in a medium sized corpus. Thus, the choice of α depends on size and diversity with respect to combinations of utility and costs within the \mathcal{P} .

Benefit-Cost Ratio The third approach to CSAL is based on the *benefit-cost ratio*, which has its roots in cost-benefit analysis frequently employed in decision theory and welfare economics. Given equal units of measurement for benefits and costs, the benefit-cost ratio (BCR) indicates whether a scenario is profitable (ratio > 1).

The BCR can be derived from the weighted product method ϕ_{WPM} applying opposing weights $\gamma_{\text{benefit}} = 1$ and $\gamma_{\text{cost}} = -1$ to combine the two conflicting criteria benefit (to be maximized) and cost (to be minimized). In contrast to the closely related net-benefit, BCR can also be applied when units are incommensurable and it is hard to find an appropriate transformation function. Although BCR applied to

incommensurable units cannot be interpreted in terms of profitability, it can still be applied to rank such examples.

This holds as long as benefit and costs can be placed in the same units by a linear transformation function. If $v_{\text{benefit}}(p) = a + \alpha \cdot u(p, \theta)$ and $v_{\text{cost}}(p) = b + \beta \cdot \text{cost}(p)$, one can refrain from finding proper values for the above variables a, α, b, β , and instead use $v_{\text{benefit}}(p) = u(p, \theta)$ and $v_{\text{cost}}(p) = \text{cost}(p)$. The BCR is defined as

$$\begin{aligned}\phi_{\text{BCR}}(p) &= v_{\text{benefit}}(p)^{\gamma_{\text{benefit}}} \cdot v_{\text{cost}}(p)^{\gamma_{\text{cost}}} \\ &= \frac{v_{\text{benefit}}(p)}{v_{\text{cost}}(p)}.\end{aligned}$$

For annotation costs, there is a linear relationship $v_{\text{cost}}(p) = b + \beta \cdot \text{cost}(p)$. The opposite, however, often holds for utility scores, especially when informativeness is estimated based on confidence scores.⁷ In consequence, the attribute-value function for benefit can often not be described as a linear function on utility.

On the assumption of a linear relationship between utility function and the respective attribute-value function, BCR has already been proposed for CSAL by Settles et al. (2008) and Haertel et al. (2008b). Our approach is an extension to their work as we explicitly consider scenarios where such a linear relationship is not given and propose a non-linear transformation function.⁸

In a direct comparison of LRK with BCR, LRK may be used when such a transformation function would be needed but is unknown. Choosing LRK over BCR is also motivated by findings in the context of data fusion in Information Retrieval, where Hsu and Taksa (2005) stated that, given incommensurable units and scales, one would do better to combine ranks rather than the actual scores or values.

CSAL based on LRK or BCR is realized by simply exchanging the utility function $u(p, \theta)$ in the NER-specific AL framework by $\phi_{\text{LRK}}(p)$ or $\phi_{\text{BCR}}(p)$, respectively.

11.4.2 Evaluation

Experimental Settings We evaluated the three approaches to CSAL, namely CCS, LRK, and BCR, on the MUC7_T corpus using the same experimental settings as in the previous section. As utility scores to estimate benefits we applied the $u_{\text{MA}}^{\bar{s}}$ and the

⁷Although normalized to $[0, 1]$, confidence estimates, especially for sequence classification, are often not on a linear scale so that confidence values that are twice as high do not necessarily mean that the utility $u(p, \theta)$, when directly derived from confidence, is also doubled.

⁸The fact that BCR may fail if utility estimates are poor and the need for a calibration of the utility score were discussed in personal communication with Robbie Haertel from Brigham Young University. Thanks a lot for your valuable comments and feedback!

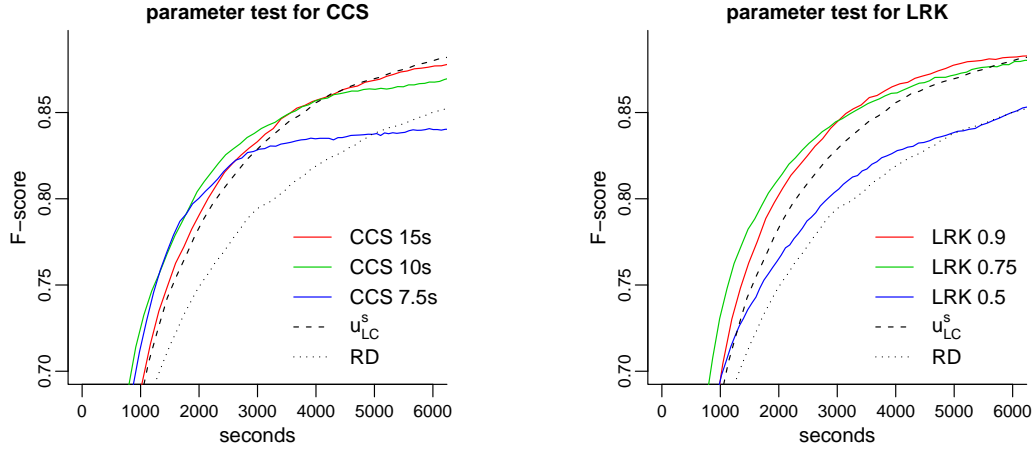


Figure 11.6: Different parameter settings for CCS and LRK.

u_{LC}^s functions. u_{MA}^s performed best in the cost-sensitive evaluation but presumably has less room for improvement compared to u_{LC}^s .

Plots show costs in terms of annotation time in seconds. Learning curves are only shown for the start-up and the transition phase. Later on in the convergence phase, due to the two conflicting criteria now considered simultaneously, selection options are extremely limited so that CSAL in those regions naturally performs sub-optimally. Again, only results for annotator A are shown.

Parametrization of CSAL Approaches Preliminary experiments were run to analyze how different parameters affect the respective CSAL approaches. For CCS and LRK, experiments were run in combination with the u_{LC}^s utility function.

For CCS, the c_{max} has to be specified. We tested three values, 7.5, 10, and 15 for c_{max} . Firstly, we tested the supervised maximum performance attainable on MUC7 $_{\mathcal{T}}$ (annotator A) when only examples below the particular c_{max} value were included. For 7.5s a maximum of $F_{max} = 0.84$ was yielded, for 10s $F_{max} = 0.86$, and for 15s $F_{max} = 0.88$. Figure 11.6 shows the learning curves of CSAL with CCS and different c_{max} values. With $c_{max} = 15$, no difference can be observed compared to cost-insensitive AL. As expected, CCS with lower values for c_{max} stagnates at the maximum performance reported above, but improves upon cost-insensitive AL in early AL iterations.

At some point all economical examples, i.e., those with costs below c_{max} but high utility, have run out. In a corpus much larger than MUC7 $_{\mathcal{T}}$ this effect will occur

with a temporal delay – with a restrictive value for c_{max} , the same exhaustion effect will occur. It is currently unclear how to specify c_{max} appropriately in a real-life annotation scenario where pretests for maximum performance for a particular c_{max} are not possible. For further experiments, we chose $c_{max} = 10$ seconds.

For LRK, we experimented with three different weights α for the utility scores including 0.5, 0.75, and 0.9. Figure 11.6 shows the effects of these weights on the learning curve. Similar tendencies as for c_{max} for CCS can be observed. With $\alpha = 0.9$, CSAL does not fall below default AL, at least in the observed range. A lower weight of $\alpha = 0.75$ results in larger improvements in earlier AL iterations but then falls back to default AL and in later AL iterations (not shown here) even below default AL. If time, however, is given too much influence with $\alpha = 0.5$, performance falls to random selection-level. This is presumably also due to corpus exhaustion. For further experiments we chose $\alpha = 0.75$ because of its potential to improve upon AL in early AL iterations.

For BCR, we specify the attribute-value function for $u_{MA}^{\bar{s}}$ as

$$v_{\text{benefit}}^{\text{MA}}(p) = n \cdot u_{MA}^{\bar{s}}$$

where n is the length of the respective sentence. This essentially leads to a summation of all token-level confidence scores u_{MA} . For u_{LC}^s we suspect a linear relationship between $v_{\text{benefit}}(p)$ and u_{LC}^s not to be appropriate; instead, a non-linear calibration function would be needed here to transform u_{LC}^s into a proper benefit estimator. This is because u_{LC}^s is based on $P_{\theta}(\vec{y}|\vec{x})$ (cf. Equation 2.24 on page 19) for confidence estimation of the complete label sequence \vec{y} and a u_{LC}^s score twice as high presumably does not indicate doubled benefit for classifier training.

To determine such a non-linear calibration function, the *true* benefit of an example p would be needed. In the absence of such information, we consider $v_{\text{benefit}}^{\text{MA}}(p)$ as a good estimation of the true benefit of an example p . To identify the relationship between u_{LC}^s and $v_{\text{benefit}}^{\text{MA}}$, we trained a model on a random subsample from $P' \subset \mathcal{P}$, then used this model to obtain the scores for u_{LC}^s and $v_{\text{benefit}}^{\text{MA}}$ for each example from the test set \mathcal{T} .⁹ Figure 11.7 shows a scatter plot of these scores and evidences that the relationship between u_{LC}^s and benefit is indeed non-linear.¹⁰

As attribute-value function for u_{LC}^s we thus propose

$$v_{\text{benefit}}^{\text{LC}}(p) = e^{\beta \cdot u_{LC}^s(p)}. \quad (11.1)$$

⁹We experimented with different sizes for P' with almost identical results.

¹⁰Pearson's correlation coefficient is relatively high ($corr = 0.6495$) because there are many examples with values for $v_{\text{benefit}}^{\text{MA}}(p)$ in $[0, 1]$ and a corresponding u_{LC}^s score in $[0.8; 0.9]$.

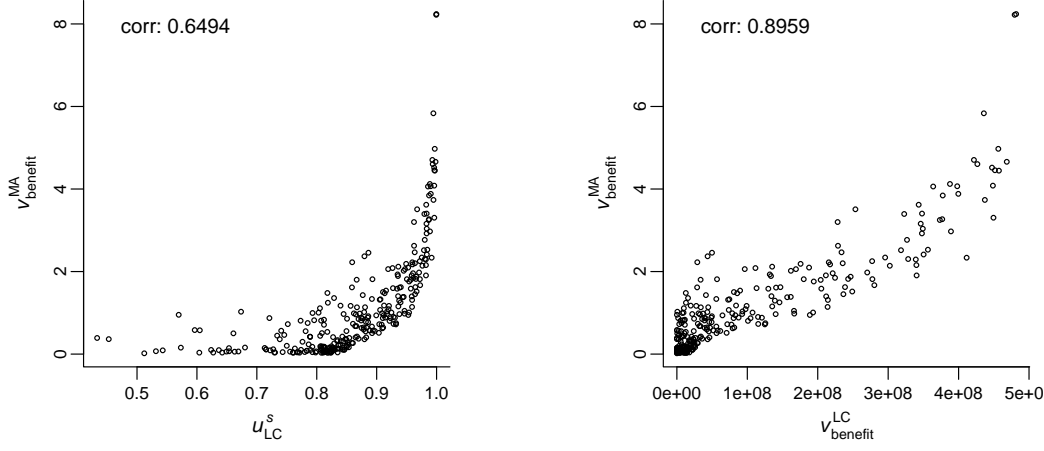


Figure 11.7: Scatter plot for u_{LC}^s versus $v_{benefit}^{MA}$ (left) and $v_{benefit}^{LC}$ versus $v_{benefit}^{MA}$ (right).

Experimentally, we determined $\beta = 20$ as a good value.¹¹ Figure 11.7 proves that $e^{\beta \cdot u_{LC}^s(p)}$ is a better estimator; the correlation coefficient is now $corr = 0.8959$.

In Figure 11.8, learning curves for BCR with $v_{benefit}^{LC}(p)$ and $v_{benefit}(p) = u_{LC}^s$ are shown. BCR with benefit based on the uncalibrated utility function fails miserably. This renders evident our hypothesis that while u_{LC}^s may be appropriate for *ranking* examples, it is inappropriate for *estimating* true benefit. BCR with $v_{benefit}^{LC}(p)$, in contrast, outperforms cost-*insensitive* AL. For further experiments with BCR, the two attribute-value functions $v_{benefit}^{MA}(p)$ and $v_{benefit}^{LC}(p)$ are applied.

Comparison of CSAL Approaches Finally, we compare all three approaches to CSAL in the parametrization chosen above for the utility function u_{MA}^s and u_{LC}^s . Resulting learning curves are shown in Figure 11.9. Improvements over cost-*insensitive* AL are only achieved in early AL iterations up to 2,500s (for CSAL based on u_{MA}^s) or 4,000s (for CSAL based on u_{LC}^s) of annotation time, respectively. This exclusiveness of early improvements can be explained as being a result of the corpus size and by this the limited number of good selection options. Since AL selects with respect to two conflicting criteria, the pool \mathcal{P} should probably be much larger to increase the repository of examples, that are favorable with respect to both criteria.

Improvements for CSAL based on u_{LC}^s are generally higher than for u_{MA}^s . Figure 11.4 already showed that in default AL with cost-sensitive evaluation, u_{MA}^s clearly outperformed u_{LC}^s . Thus, u_{LC}^s probably offers more room for improvement. Moreover,

¹¹We determined β in a rather ad-hoc manner. Future work may concentrate on better fits.

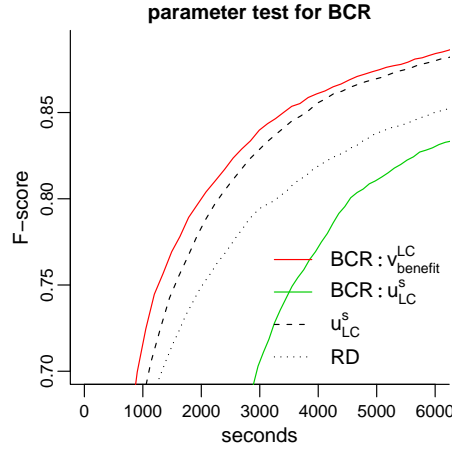


Figure 11.8: Different parameter settings for BCR.

cost-*insensitive* AL based on u_{LC}^s does not exhibit any normalization where, in contrast, $u_{\text{MA}}^{\bar{s}}$ is normalized at least to the number of tokens in an example. Now, in CSAL, both u_{LC}^s and $u_{\text{MA}}^{\bar{s}}$ are normalized by costs (in different ways according to the respective CSAL approaches), which amounts to a more substantial methodological enhancement for u_{LC}^s than for $u_{\text{MA}}^{\bar{s}}$.

For CSAL based on $u_{\text{MA}}^{\bar{s}}$ it is hard to distinguish a clear winner among the different approaches. However, all three CSAL approaches improve upon cost-*insensitive* AL. For CSAL based on u_{LC}^s , LRK performs best, while CCS and BCR perform similarly well. Given this result, we might prefer LRK or CCS over BCR. A disadvantage of these two approaches is that they require corpus-specific parameters. Appropriate parametrization may be difficult for a new learning problem for which no data for experimentation is available. Even though it does not exhibit the best performance, BCR does not require further parametrization and appears more appropriate for real-life annotation projects – as long as utility is an appropriate estimator for benefit.

Previously, CSAL with BCR had been applied by Settles et al. (2008). The authors also applied a utility function based on sequence-confidence estimation, which presumably, as with our u_{LC}^s utility function, is not a good estimator for benefit. The fact that Settles et al. did not explicitly treat this issue might explain why CSAL with BCR was often worse than cost-*insensitive* AL in their experiments.

CSAL applied to SeSAL Finally, we experimented with a cost-sensitive version of SeSAL. SeSAL was changed slightly so as to incorporate cost based on the BCR

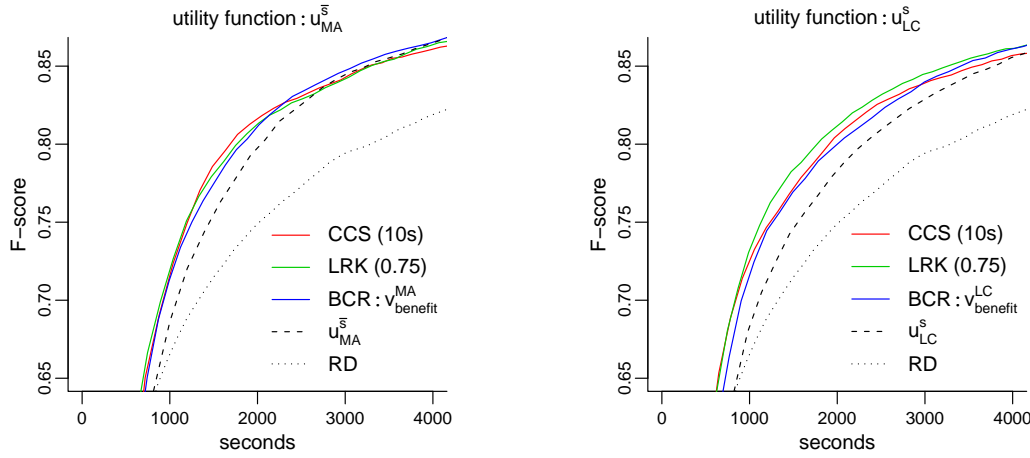


Figure 11.9: Comparison of CSAL approaches for the utility functions u_{MA}^s and u_{LC}^s . Baseline given by random selection (RD) and standard AL with either u_{MA}^s or u_{LC}^s .

method, based on the $v_{benefit}^{LC}(p)$ attribute-value function. Figure 11.10 shows learning curves for cost-*insensitive* and cost-sensitive SeSAL and FuSAL and reveals that cost-sensitive SeSAL considerably outperforms cost-sensitive FuSAL. Cost-sensitive SeSAL attains a target performance of $F=0.85$ with only 2806s, cost-sensitive FuSAL needs 3410s, and random selection consumes over 6060s. Thus, cost-sensitive SeSAL here reduces true annotation time by about 54 % compared to random selection, while cost-sensitive FuSAL reduces annotation time by only 44 %.

11.4.3 Conclusions

Overall, this section showed that improvement in terms of actual annotation time upon standard AL can indeed be achieved by making AL cost-sensitive. As for fully-supervised AL, its standard, cost-*insensitive* variant already exhibits an improvement over random selection in terms of annotation times. However, after cost-sensitization by any of our three proposed approaches, it performed even better. The cost-*insensitive* variant of SeSAL is no better than FuSAL when evaluated against time. However, in its cost-sensitized variant, SeSAL outperforms FuSAL.

11.5 Summary

This chapter has studied several aspects of CSAL. With the MUC7 \mathcal{T} corpus a precious linguistic resource needed for simulations of AL in a cost-realistic environment

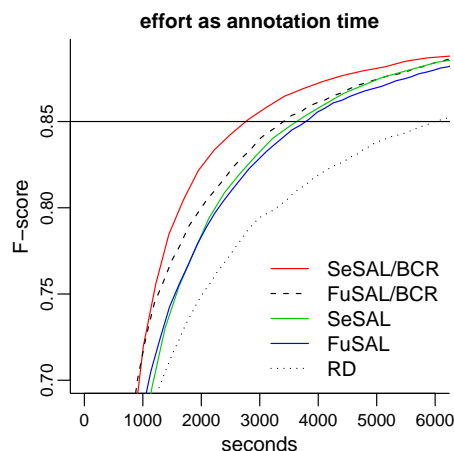


Figure 11.10: Evaluation of the cost-sensitive version SeSAL. Cost-sensitivity is based on the BCR method. u_{LC}^s is used as utility function for SeSAL and FuSAL.

has been created. With information on the time it takes to add certain linguistic annotations, such as NER in our case, the MUC7 \mathcal{T} corpus introduces a new breed of meta-information for a linguistic corpus. This resource is unique in its size and its level of annotation- and time measurement-granularity (sentences and CNPs).

With MUC7 \mathcal{T} , we were able to evaluate our approaches to both fully- and semi-supervised AL (cf. Chapters 4 and 6) in a more realistic scenario, with real annotation costs. Moreover, based on these results, several approaches were proposed to cost-sensitize the fully- and semi-supervised approaches to AL. Overall, our evaluation of cost-*ins*sensitive and cost-sensitive approaches to AL provide empirical evidence that AL can indeed considerably reduce annotation effort in terms of time needed for annotation. Moreover, approaches to AL are even more efficient in their cost-sensitive variant. Such findings are good news for potential users of AL many of whom are skeptical about the real monetary benefit of AL.

As an another outcome, our experiments also revealed that the pool size for CSAL needs to be exceptionally large to provide a good repository of examples properly covering the two conflicting criteria benefit and cost. As for MUC7 \mathcal{T} , we found its size of 3,113 sentences to be not quite large enough. This renders the evaluation of approaches to CSAL problematic and suggests interpretation of the results only for early AL iterations. MUC7 \mathcal{T} may, however, still serve as an empirical foundation for deriving annotation cost models allowing the prediction of annotation time for unlabeled examples. Predictive cost models are essential for the realization of CSAL in real-life annotation projects where information about annotation time has to be predicted in advance of actual annotation.

Part IV

Active Learning in Practice

Chapter 12

Environment for Active Learning-Driven Annotation

Due to the enormous need for Named Entity annotations caused by the STEMNET research project (see Section 12.2 for details), an environment for rapid Named Entity annotation has been developed as part of this thesis. As one of its core features, the Jena ANnotation Environment (JANE), allows for AL-based annotation. JANE supports the whole annotation life-cycle including the compilation of annotation projects, annotation itself, monitoring, and the deployment of annotated material. While originally developed for Named Entity annotation, JANE is based on a modular architecture so that it can be adapted to new annotation tasks.

This chapter presents JANE, motivates its functionalities from a practical point of view, and discusses its system architecture. This work has been published before in Tomanek et al. (2007b). Subsequently, the application of JANE within the STEMNET project is described.

12.1 System Architecture

JANE has a distributed system architecture. It consists of four components. The annotation repository is the central component where all relevant data is stored and internal states are hold. There are two clients, one for annotation and one for administration. Finally, the AL component handles all AL-related processes. All components communicate with the annotation repository through a network socket. Figure 12.1 shows JANE's system architecture; bold arrows symbolize direct communication, dashed arrows indirect communication. JANE is largely platform-independent because all components are implemented in Java.

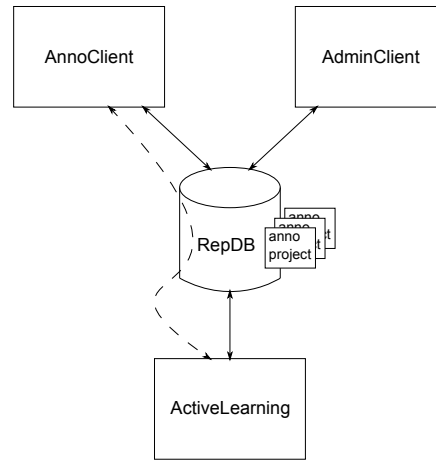


Figure 12.1: System architecture of JANE; bold arrows indicate direct communication, dashed ones indirect communication.

An *annotation project* consists of a collection of documents, an associated annotation schema – a specification of what has to be annotated in which way – a set of configuration parameters, and an annotator assigned to it.

Two types of annotation projects are distinguished: A *default project*, contains a predefined and fixed collection of documents. In such a project, all documents are annotated independently of each other. In an *AL project*, the annotator has access to a single document at a time. Having annotated this document, AL selection is run and a new document is dynamically compiled from the actively selected sentences.

12.1.1 Administration Client

Administration of large-scale annotation endeavours is a challenging management task. Figure 12.2 shows a snapshot of the user interface of the administration client. The administration client supports many relevant subtasks:

User Management Create and manage accounts of annotators.

Creation of Projects The creation of an annotation project requires a considerable number of documents and other files such as annotation schema definitions to be uploaded to the annotation repository. Furthermore, several configuration parameters can be specified including for example the number of sentences to select in each AL iteration.

Editing a Project The administrator can reset a project, i.e., delete all actual annotations made to the project, but keep the general set-up of the project

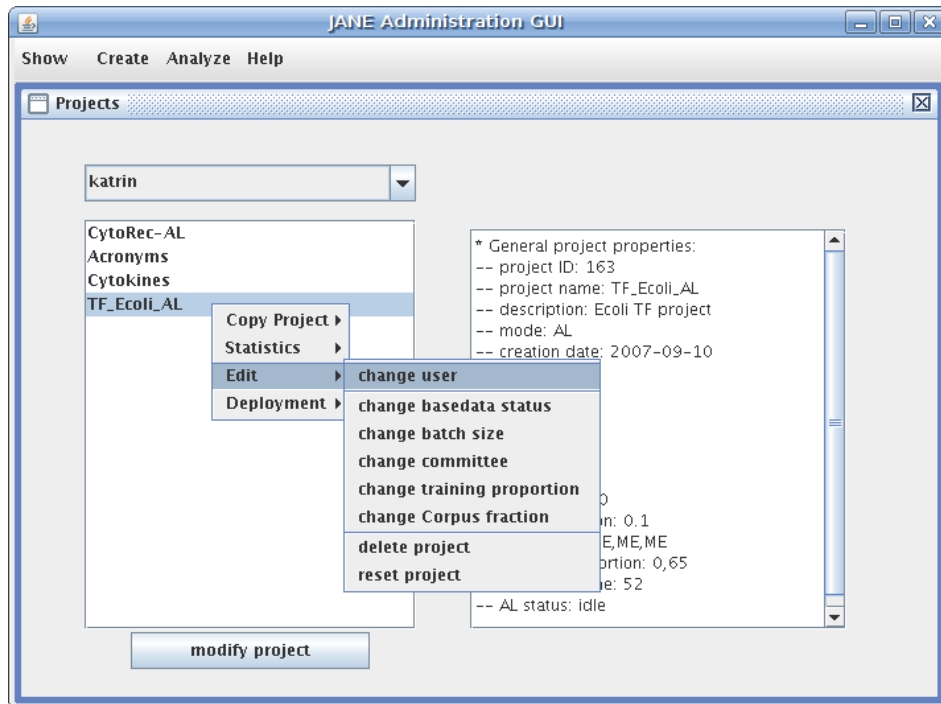


Figure 12.2: Graphical user interface of administration client.

unchanged. This is especially helpful in a pre-testing phase when annotation guidelines change. Moreover projects can be copied (including all annotations made) which is helpful when several annotators label the same documents to check the applicability of the guidelines by inter-annotator agreement calculation. Modification of several project-specific parameters and files as well as the complete deletion of the project is also possible.

Monitoring the Annotation Process The administrator can check which documents of an annotation project have already been annotated, how long annotation took on the average, etc. Furthermore, the progress of AL projects can be visualized by learning and disagreement curves (cf. Chapter 5).

Inter-Annotator Agreement For related projects (projects sharing the same annotation schema) the degree to which several annotators mutually agree in their annotations can be calculated. Such inter-annotator agreement (IAA) is a common estimate of the quality and applicability of particular annotation guidelines (Kim and Tsujii, 2006). Currently, several IAA metrics of different strictness are available.

Deployment The annotation repository stores the annotations in a specific XML

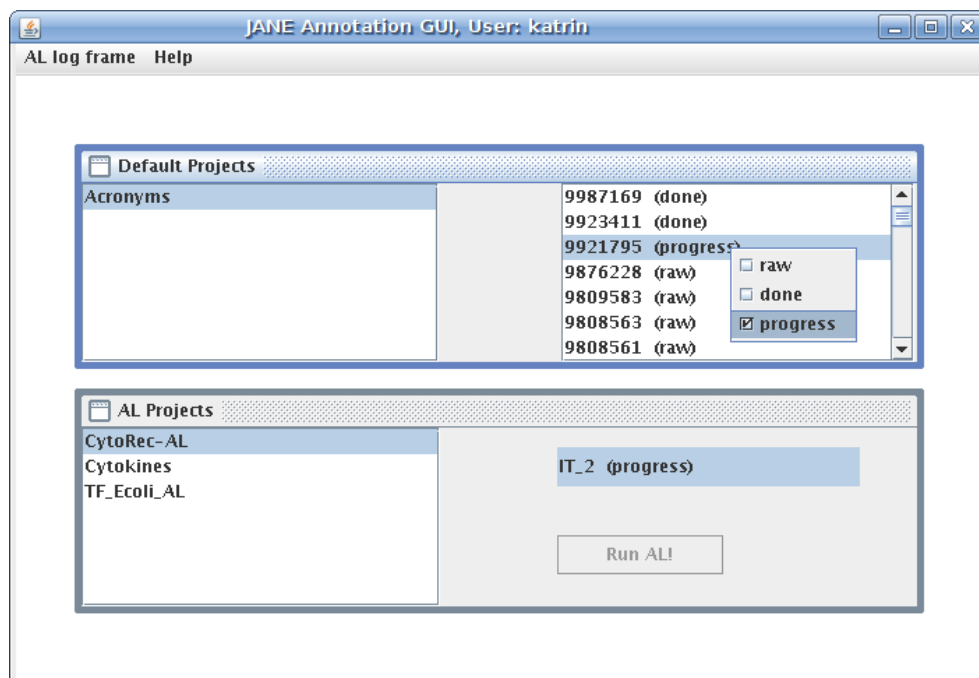


Figure 12.3: Graphical user interface of the annotation client.

format. Deployment allows to export the annotations in different output formats. Both for IAA calculation as well as deployment, only documents marked as *completely annotated* by the annotators are considered.

12.1.2 Annotation Client

As the annotators are domain experts – for example graduate students of biology in the STEMNET project – rather than computer specialists, we wanted to make life for them as easy as possible. Especially we did not want them to deal with annotation files manually. All actions necessary in context of annotation work can be done with the annotation client. The annotation client is in charge of all the communication with the annotation repository (see Figure 12.3). It provides access to an external editor for the actual annotation process.

After log-in to the annotation client, the annotator sees a list of her annotation projects along with a short description. Double clicking on a project, she receives a list with all documents in this project. Documents have different flags (*raw*, *in progress*, *done*) to indicate the current annotation state as set by the annotator.

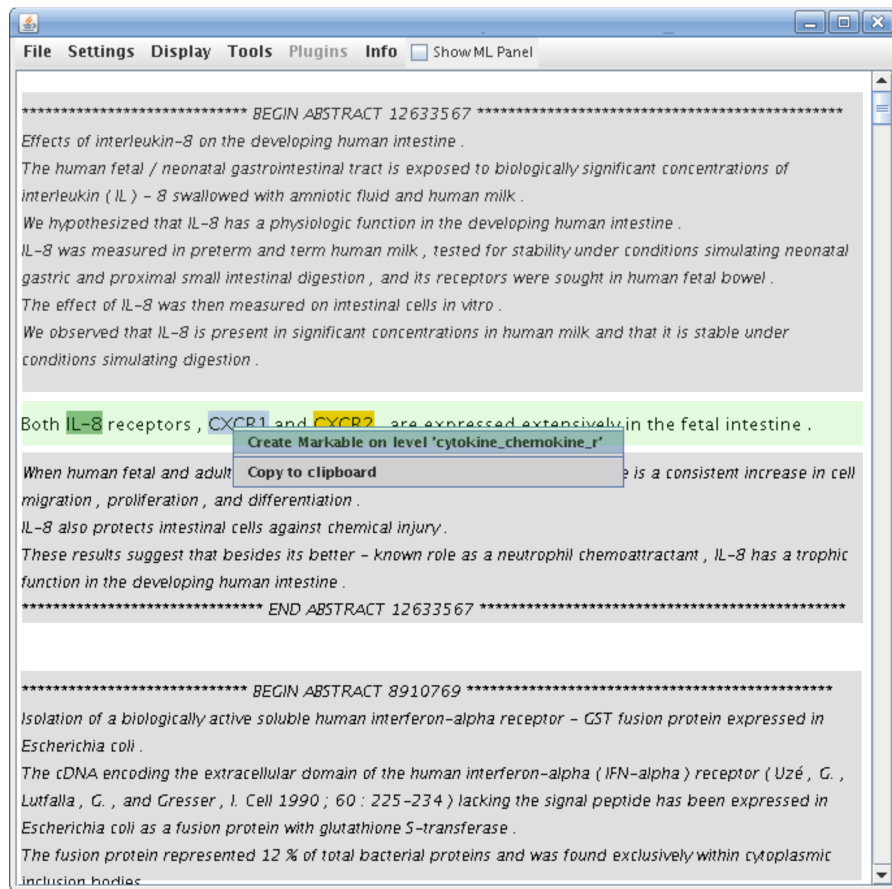


Figure 12.4: AL-enabled annotation in MMAX2 editor. Gray shaded areas are context information and cannot be annotated. Green indicates that this should be annotated.

Moreover, a comment can be added for each document. This facility proved useful in larger annotation projects, where annotators frequently had problems with the guidelines and could communicate this to the administrator through comments.

Annotation itself is done with MMAX2, an external annotation editor developed by Müller and Strube (2003).¹ While there are several annotation editors available we decided for MMAX2 as it can be flexibly configured for the specific annotation task and supports multi-level as well as chained annotations, which are important for other annotation tasks in context of IE. The document to be annotated, the actual

¹MMAX2 is now an open-source project (<http://mmax2.sourceforge.net/>). We made several modifications to it for better convenience such as, for example, optimized color highlighting of annotations.

annotations, and the configuration parameters are stored in MMAX2-specific XML files. The annotation repository reflects this data structure. The administration client provides several export functionalities so that the MMAX2-specific data is converted into other, de-facto annotation formats.

Double clicking on a specific document from the annotation GUI directly opens an MMAX2 frame for annotation (see Figure 12.4). When working on an AL project, the annotator can start the AL selection process (which then runs on a separate high-performance machine) once having finished the annotation of the current document. During the AL selection process, which may take up to several minutes according to the particular configurations, the current project is blocked. However, meanwhile the annotator can continue to work on other annotation projects.

For AL purposes, the sentences selected in one AL iteration, are compiled into a single document. Besides the selected sentences, such a document also contains the context surrounding each sentence. However, this context cannot be annotated and is shown in italics in MMAX2. Additionally, an identifier of the original source document from which a sentence was taken is shown. This proved beneficial in scenarios where ambiguity could not be resolved with the local context so that the annotators had to refer to the full texts to be able to make an informed decision. Figure 12.4 shows the selected sentences and the surrounding context.

12.1.3 Annotation Repository

The annotation repository is the heart of the annotation environment. All project, user, and annotation relevant data is stored there. This central data management is crucial for backup and deployment activities. Additionally, annotators do not have to care about how to shift the annotated documents to the managerial staff. All state information related to the entire annotation cycle is recorded in the repository.

The repository is implemented as a relational database.² The components communicate with each other only through the annotation repository. In particular, there is no direct communication between the annotation client and the AL component. Each component can be run on a different machine as long as it has a network connection to the annotation repository. This has two major advantages: Firstly, annotators can work remotely (e.g., from home or from a physically dislocated lab or from at home). Secondly, resource-intensive tasks such as the AL-based selection of sentences can be run on separate, high-performant machines.

²We chose MySQL, a fast and reliable open source database with native Java driver support through the JDBC network driver (<http://java.sun.com/javase/technologies/database/>).

12.1.4 Active Learning Component

The AL component is currently based on the QbB-based AL and employs the Vote Entropy utility function u_{VE}^s . Complete sentences are selected. The selected sentences are shown to the annotator in their surrounding context for convenience. The composition of the QbB committee can be chosen in the administration client allowing both for homogeneous and heterogeneous committees.

According to our findings on sample reusability on the NER tasks (cf. Chapter 7), in our real-life annotation projects we often employ a MaxEnt learner during the AL selection for a final CRF-based consumer to reduce computational complexity and to cut down annotator idle time. In the same line, instead of actively selecting from the complete pool of unlabeled data which in practise often subsumes several millions of sentences, selection can be based on a random subsample of the pool.

12.2 Large-Scale Annotation of Biomedical Documents

Information extraction (IE) from biomedical documents is a lively field of research. As other domains away from the standard, widely covered news-paper domain, IE in the biomedical domain still has a shortcoming of annotated corpora. The creation of new annotations is inevitable when porting (semi-) supervised methods to new fields, domains, or genres. Moreover, annotation is necessary because different sets of entity types are relevant in other domains. As shown throughout this thesis, AL has the potential to reduce the effort needed to accomplish this enormous annotation task. And given such a methodology, (sub)domain changes now seem more feasible without committing to overly excessive annotation costs.

JANE was applied to serve the need for large-scale Named Entity annotations required by STEMNET, a research project where IE methods were applied to documents from the field of hematopoietic stem cell transplantation.³ One major goal of STEMNET was to build a broad-coverage semantic search engine targeting at researchers from the life sciences and especially the biomedical domain. This search engine, called Semedico, provides deep semantic access to the contents of PUBMED abstracts.⁴ All semantic meta-data accessible through Semedico was automatically extracted by means of NLP which basically involved recognition of a large amount of different entity types.

³STEMNET ran from 2006 to 2009 and was funded by the German Federal Ministry of Education and Research (BMBF).

⁴<http://www.semedico.org>

entity category	distinct entity types	tokens annotated
cytokines and growth factors	7	276,570
cytokine and growth factor receptors	7	223,314
antigens (e.g. CD antigens)	6	196,063
minor histocompatibility antigens	6	187,496
organisms and organism attributes	18	173,943
T cells and natural killer cells	15	158,856
B cells and dendritic cells	14	164,790
hematopoietic progenitor cells	10	146,482
genomic variations	6	139,961
Σ	89	1,667,475

Table 12.1: Large-scale and entity-rich semantic annotations for the STEMNET project.

As a preparatory step for the annotation phase, a large collection of scientific abstracts for each entity category was selected from PUBMED using keyword queries. The sentences of these abstracts make up the pool of unlabeled examples from which AL selects. Since the pools often consisted of over 1 million sentences, in every AL iteration a random subsample of about 40,000 sentences was drawn from which AL then selected.

Table 12.1 gives an overview of the semantic types and amount of entities being annotated. We annotated 89 distinct entity types relevant for immunogenetics such as various protein types (cytokines, growth factors and their receptors, major and minor histocompatibility antigens, transcription regulators), chemicals (ligands), various kinds of immune cells (various sorts of T-, B-, NK- and dendritic cells), blood progenitor cells, and organisms. More details on the annotation endeavor, including guidelines as well as inter-annotator statistics, are reported in Hahn et al. (2008).

12.3 Conclusions

The availability of an AL-enabled annotation environment is among the core requirements to allow NLP practitioners to benefit from the potential savings in annotation effort through AL. This issue is also addressed in the next Chapter. While JANE is currently not publicly available, it could serve as a first prototype when developing such a publicly available annotation framework.

Chapter 13

Survey on the Practical Usage of Active Learning

In the previous chapters we have discussed various aspects of AL. AL has been successfully tested on a range of NLP tasks including NER, as in this thesis. However, despite impressive results in terms of reduced annotation effort reported by the various studies on AL for NLP, it seems that AL is still not applied as a standard annotation technique for real-life annotation endeavors. This chapter presents the results of a survey arranged to analyze the extent to which AL is used to support the annotation of textual data in practice. Moreover, it also aims at addressing the reasons to why or why not AL has been found applicable to a specific task.

At the time of writing, there were only a few works investigating into the real usage of AL through surveys or user studies. Most of the studies were concerned with the question of cost-aspects in context of AL (see Section 11.1 for a discussion of previous work in that field). Other user studies include that of Palmer et al. (2009) on AL for language documentation, and the work of Ngai and Yarowsky (2000) comparing manual rule writing and AL-based annotation to construct a noun phrase chunker.

This survey serves as an indicator about the maturity of this field of research for practical application, and gives hints for aspects considered relevant by practitioners, insufficiently studied or completely overlooked so far. This work has been published in (Tomanek and Olsson, 2009).

13.1 Survey Set-Up

The survey was realized in the form of a web-based questionnaire. The survey targeted participants who were involved in the annotation of textual data intended for machine learning for all kinds of NLP tasks. The call for participation in the

survey was sent to a number of mailing list relevant in the field of NLP and ML.¹ Utilizing these mailing lists, we expect to have reached a fairly large portion of the researchers likely to be involved in annotation projects for NLP. The questionnaire was open February 6–23, 2009.

After an introductory description and one initial question, the questionnaire was divided into two branches. The first branch was answered by those who had used AL to support their annotation, while the second branch was answered by those who had not. Both branches shared a common first part about the general set-up of the annotation project under scrutiny. The second part of the AL-branch focused on experiences made with applied AL. The second part of the non AL-branch asked questions about the reasons why AL had not been used. Finally, the questionnaire was concluded by a series of questions targeting the background of the participant.

13.2 Questions and Answers

147 people participated in the survey. 54 completed the survey while 93 did not which constitutes an overall completion rate was 37 %. Most of the people who did not complete the questionnaire answered the first couple of questions but did not continue. Their answers are not part of the discussion below. We refrain from a statistical analysis of the data but rather report on the distribution of the answers received.

Of the people that finished the survey, the majority (85 %) came from academia, with the rest uniformly split between governmental organizations and industry. The educational background of the participants were mainly computational linguistics (46 %), general linguistics (22 %), and computer science (22 %).

Questions Common to both Branches Both the AL and the non-AL branch were asked several questions about the set-up of the annotation project under scrutiny. The questions concerned whether AL had been used to support the annotation process, the NLP tasks addressed, the size of the project, the constitution of the corpus annotated, and how it was decided when to stop the annotation process.

¹Mailing lists include: the BioNLP mailing list (<http://bionlp.org/>), Corpora list (<http://gandalf.aksis.uib.no/corpora/>), UAI list (http://www.urantia-uai.org/UAI_List.html), ML-news (<http://groups.google.com/group/ML-news>), SIG-IRList (<http://www.sigir.org/sigirlist/>), Linguist list (<http://linguistlist.org/>), and the member lists of SIGANN (<http://www.cs.vassar.edu/sigann/>), SIGNLL (<http://ifarm.nl/signll/>), and ELRA (<http://www.elra.info/>).

1. Use of AL as annotation support The first question posed was whether people had used AL as support in their annotation projects. 11 participants (20 %) answered this question positively, while 43 (80 %) said that they had not used AL. Depending on the answer to this question, the participant was then asked questions from either the AL branch of the questionnaire, or from the non-AL branch.

2. Task addressed Most AL-based annotation projects concerned information extraction (IE) (52 %), document classification (17.6 %), and (word sense) disambiguation (17.6 %). Also in non AL-based projects, most participants had focused on IE tasks (36.8 %). Here, syntactic tasks including part-of-speech tagging, shallow, and deep parsing were also often considered (19.7 %). Textual phenomena, such as co-references and discourse structure (9.6 %), and word sense disambiguation (5.5 %) formed two other answer groups. Overall, the non AL-based annotation projects covered a wider variety of NLP tasks than the AL-based ones. All AL-based annotation projects concerned English texts. In contrast, 37.2 % of the non-AL projects dealt with texts in other languages.

3. Size of the project The participants were also asked for the size of the annotation project in terms of number of units annotated, number of annotators involved and person months per annotator. The average number of person months spent on non AL-projects was 21.2 and 8.7 for AL-projects. However, these numbers are subject to a high variance and the difference is not statistically significant ($p = 0.205$ in a Student' t-test). Yet overall, we had the impression that non AL-based projects were larger in terms of annotators involved, number of person months spent, complexity of annotation task addressed, etc.

4. Constitution of the corpus Further, the participants were asked how the corpus of unlabeled examples was selected. This examples are used as the pool to select from when using AL, and as the corpus to be annotated in non AL-based annotation. The answer options included (a) taking all available instances, (b) a random subset of them, (c) a subset based on keywords, and (d) others. In the AL-branch, the answers were uniformly distributed among the alternatives. In the non AL-branch, the majority of participants had used alternatives (a) (39.5 %) and (b) (34.9 %).

5. Decision to stop the annotation process A last question regarding general annotation project execution concerned the stopping of the annotation process. In AL-based projects, evaluation on a held-out test set (36.5 %) and the exhaustion of money or time (36.5 %) were the major stopping criteria. An AL-specific stopping criterion were used only once, while in two cases the annotation was stopped because the expected gains in model performance fell below a given threshold. In almost half (47.7 %) of the non AL-based projects the annotation was stopped since the available

money or time had been used up. Another major stopping criterion was the fact that the complete corpus was annotated (36 %). Only in two cases annotation was stopped based on an evaluation of the model achievable from the corpus. These results are quite impressive as this shows that for both AL and non-AL projects, the use of sophisticated stopping criteria is not established, yet.

Questions Specific to the AL-Branch The AL-specific branch of the questionnaire was concerned with two aspects: the learning algorithms involved, and the experiences of the participants regarding the use of AL as annotation support. Percentages presented below are all related to the 11 persons who answered this branch.

1. Learning algorithms used As for the AL methods applied, there was no single most preferred approach. 27.3 % had used uncertainty sampling, 18.2 % query-by-committee, another 18.2% error reduction-based approaches, and 36.4 % had used an “uncanonical” or totally different approach which was not covered by any of these categories. As learning algorithms for the selectors, maximum-entropy based approaches as well as SVMs were most frequently used (36.4 % each).

2. Experiences When asked about their experiences, the participants reported that their expectations with respect to AL had been partially (54.4 %) or fully (36.3 %) met, while one of the participants was disappointed. Unfortunately, the participants from the AL-branch did not leave many experience reports in the free text field. From the few received, it was evident that the computational complexity of AL and the resulting delay or idle time of the annotators, as well as the interface design are critical issues in the practical realization of AL as annotation support. Two comments especially support these claims:

”We could not perform anything other than batch-mode labeling because of the computational complexity of ALL incremental AL methods.”

”Never been able to build an AL interface which was easy to convince annotators to use it.”

Question Specific to the Non-AL Branch The non AL-specific branch of the questionnaire was basically concerned with why people did not use AL as annotation support and whether this situation could be changed. The percentages given below are related to the 43 people who answered this particular part of the questionnaire.

1. Why was AL not used? Participants could give multiple answers to this question. Many participants had either never heard of AL (11 %) or did not use AL due to insufficient knowledge or expertise (26 %). The implementational overhead to develop an AL-enabled annotation editor kept 17.8 % of the participants from using

AL. Another 19.2 % of the participants stated that their project specific requirements did not allow them to use AL. Given the comments given in the free text field, it can be deduced that this was often the case when people wanted to create a corpus that could be used for a multitude of purposes (such as building statistics, cross-validation, learning about the annotation task per se, and so forth) and not just for classifier training. In such scenarios, the sample selection bias introduced by AL is certainly disadvantageous. Finally, about 20.5 % of the participants were not convinced that AL would work well in their scenario or really reduce annotation effort. Some participants stated in their free form comments that while they believed AL would reduce the amount of examples to be annotated it would probably not reduce the overall annotation time.

2. Would you consider using AL in future projects? According to the answers of another question of the survey, 40 % would in general use AL, while 56 % were sceptical but stated that they would possibly use a technique such as AL. We interpret this as a general openness of the annotation community to adapt new techniques which is motivating for AL researchers.

13.3 Discussion and Conclusions

Although it cannot be claimed that the data collected in this survey is representative for the NLP community as a whole and the number of participants was too low to draw statistically firm conclusions, some interesting trends have indeed been discovered within the data itself. The conclusions drawn in this section are related to the answers provided in light of the questions posed in the survey.

The doubts towards AL as a potential aid in annotation in essence boil down to the absence of a (publicly available) AL-enabled annotation editor, as well as the difficulty in estimating the effective reduction in effort (such as time, money, labor) that the use of AL implies. Put simply: Can AL for NLP really cut annotation costs? How can AL for NLP be practically realized without too much overhead in terms of implementation and education of the annotator?

Research addressing the first question is ongoing which as evidenced, e.g., by the recent Workshop on Cost-Sensitive Learning held in conjunction with the Neural Information Processing Systems Conference (NIPS) 2008. Moreover, our study of cost-sensitive evaluation of AL and cost-sensitive AL in Chapter 11 has shown that the question whether savings in terms of corpus size translate to annotation time or cost is indeed crucial.

As for the latter question, there is evidently a need of a general framework for AL in which (specialized) annotation editors can be used. To date, there is no such software publicly available. In the previous chapter, our AL-enabled annotation environment has been discussed which might serve as a prototype for further development.

Also, hand-in-hand with the theoretical aspects of AL and their practical realizations in terms of available software packages, there clearly is a need for usage and user studies concerning the effort required by human annotators operating under AL-based data selection schemes in real annotation tasks. The first Workshop on Active Learning for Natural Language Processing (2009) already attracted some submissions on user studies in the field of AL. The sequel workshop, to be hold in 2010, has taken up this issue in its call for papers.²

Among the participants of the survey turn-around time and consequently the idle time of the annotator has been said to be a critical issue. In the same spirit, Requirement 2 has been formulated. Moreover, in this thesis AL approaches of low sampling complexity have been preferred over more complex ones which are presumably more efficient in reducing sampling complexity (cf. Chapter 4). Such considerations are especially important in context of NLP learning problems which tend to be complex due to their high-dimensional representation.

Interestingly, English was the only language addressed in AL-based annotation projects. This is somewhat surprising given that AL seems to be a technique well suited for bootstrapping language resources for so called “under-resourced” languages. The recently published work by Palmer et al. (2009) on AL for language documentation of the Mayan language Uspanteko underlines this assumption. Annotation in context of an endangered language as is Uspanteko is especially costly due to the extremely limited number of speakers.

We were also surprised by the fact that both in AL and non-AL projects rather unsophisticated stopping criteria were used. The need for proper monitoring and stopping methods for AL has been discussed in this thesis in detail in Chapter 5. While stopping criteria are especially relevant in AL-driven annotation projects to finally cash in the savings in annotation effort, the principle intuition on stopping criteria also holds for annotation scenarios where the corpus is assembled by a random selection of documents.

Overall, the main result of our survey is that the general acceptance of AL as a widely applied annotation method strongly depends on several end-user specific needs.

²The author of this thesis has been a co-organizer and co-founder of the 2009 workshop, and is also a co-organizer of the 2010 workshop. The follow-up workshop will be held at the 2010 Conference of the North American Chapter of the Association for Computational Linguistics.

Part V

Conclusions

Chapter 14

Summary and Perspectives

This thesis aimed at clarifying the theoretical background of and providing a framework for resource-aware annotation for the rapid and economical creation of labeled training data. The availability of sufficient quantities of high-quality training data is essential in order to train statistical methods for NLP tasks. These are methods that we wish to apply to address the challenges of the information era, i.e., managing the vast body of unstructured information with the goal of extracting valuable knowledge from it. The precious resource in the generation of training data is human labor because the arduous task of annotation has to be performed manually by human annotators.

Active learning (AL) is a general technique to reduce sampling complexity, i.e., the number of examples necessary to yield a model with a particular target performance. In contrast to the standard, passive learning scenario, this is achieved by giving the learner control over the data, so that there is selective sampling of those examples that are likely to be very useful for improving a model.

This thesis mainly focused on AL as a strategy for annotation of textual documents in real-world scenarios. These scenarios pose several requirements for a resource-aware annotation strategy to be practically applicable and considered efficient. Eight such requirements were formulated in the introduction to this thesis. The following section briefly summarizes each requirement and the respective contributions and key achievements made.

14.1 Summary of Contributions and Achievements

Requirement 1 postulates that a resource-aware annotation strategy must yield practically relevant savings as compared to the standard annotation strategy, in which the examples to be annotated are simply randomly selected. In Chapter 4 we presented an adaptation of the general AL framework to sequential learning problems

as they are frequently found in NLP tasks. The NLP task of Named Entity Recognition (NER) was used throughout the thesis as a sample scenario for linguistic annotation. Our NER-specific AL framework was evaluated with different utility functions on four corpora annotated with named entity mentions.

For attaining a particular target performance, we recorded possible savings of up to 80 % in annotation effort by AL compared to random sampling. Annotation effort was here measured by the number of tokens to be annotated. In terms of true annotation time, we could still record savings of up to 54 % (Chapter 11). These impressive savings were achieved using our novel approach to semi-supervised AL (Chapter 6), which requires only a few sub-sequences of the selected sentences to be manually annotated – the remainder is automatically labeled without any human intervention.

While such savings are not the exponential savings that AL can achieve in theory (cf. Section 3.4), they are highly relevant and can considerably reduce overall costs of training data creation. Most interestingly, such savings were achieved even though our NER-specific AL framework is based on heuristics to AL (Uncertainty Sampling and Query-by-Committee) instead of statistically optimal approaches, which might have reduced annotation effort even further.

The design decision for the application of such heuristics was deliberately made in accordance with Requirement 2 postulating rapid selection. In our experiments, only one (in the case of Uncertainty Sampling) or three models (in the case of Query-by-Committee) had to be trained for each AL iteration – much fewer than in the statistically optimal approaches to AL. In Chapter 7, we showed that computational complexity of selection could be reduced even further when a less complex model is used during AL selection. This additional reduction of selection time was accompanied by only a slight reduction of sampling efficiency.

Our NER-specific AL framework does not require many settings to be made – the only parameters that have to be set are the utility function, the batch size, and the size of the seed set. We successfully applied our AL framework on NER to four different corpora with identical settings. Moreover, the framework can be used flexibly with arbitrary models, which are treated in a black-box fashion. This satisfactorily meets Requirement 3, postulating a possibly generic approach flexible enough to be applied to a broad range of annotation scenarios without major modifications.

In the same spirit, in Chapter 5 we proposed a stopping criterion, which does not require any user-defined parameters to be set but finds a reasonable stopping point based only on characteristics of the data set at hand. Addressing Requirement 4, which postulates means to monitor and eventually stop the AL-based annotation process, this chapter also presented a novel approach to approximating the learning

curve progression that makes it possible to identify the three stages of learning curves (start-up, transition, and convergence stage). The learning curve is approximated in an unsupervised manner, not requiring any additional annotation effort. We showed that, based on this curve approximation, good stopping points can be found both in simulations as well as in real-world annotation scenarios. Additional research in this direction might focus on an extension of our approach so that the absolute performance levels can be derived from the approximated curve. Given such an approach, one could identify precisely the optimal stopping point for a user-defined trade-off between cost and benefits of annotation.

Sample reusability, as postulated by Requirement 5, is an important issue as it allows the created training data to be flexibly used to learn different types of models. This may be desirable when the final model to be learned in advance of data acquisition, which is usually the case in all real-world annotation endeavors. Moreover, this allows us to employ models with lower training complexity, making AL selection more rapid. In Chapter 7, the issue of sample reusability was defined as a problem of learning under sample selection bias. This chapter also reported on our large-scale evaluation of our AL framework with respect to sample reusability in scenarios where different models are used during selection and as final consumers.

While in the NER scenario, reusability was given in all realistic scenarios, no general pattern on reusability on arbitrary data sets could be found, leading us to the conclusion that reusability is to a large extent dependent on the combination of learning problem, data set, and model. Another interesting finding is that self-selection did not always constitute the upper bound for sampling efficiency of foreign-selection. While this thesis does not provide a general means of predicting the presence or degree of reusability, this chapter should be understood as a starting point for further investigation into the challenging issue of AL sample reusability.

A related issue is that of sampling data for multiple tasks, for which training data should be created simultaneously. This means that examples must be useful for all tasks involved. In Chapter 9, with reference to Requirement 6, we gave a definition of this new field of research within AL, which we called multi-task AL in contrast to the standard, single-task AL scenario. Furthermore, as a proof-of-concept, two approaches to multi-task AL were presented and evaluated in a scenario where training data were to be provided for two highly dissimilar tasks. Results gave encouraging evidence that the consideration of multiple tasks during AL-based selection of training data is possible and that it can indeed reduce annotation effort even further.

According to the Zipfian nature of natural language, class imbalance is a ubiquitous problem in NLP. Fulfilling Requirement 7, in Chapter 10 we proposed several novel approaches to re-balancing skewed data from the start during AL-driven data

acquisition, instead of as a post-processing step, as is usually done in machine learning literature. Our evaluation showed that class imbalance can indeed be reduced effectively as early as during data acquisition time and thus increase classifier performance without additional annotation cost.

Finally, one of the most crucial aspects of a resource-aware annotation strategy is that it actually reduces the *true* annotation effort, as postulated by Requirement 8. This issue was studied in detail in Chapter 11. As a prerequisite for experiments on cost-sensitive AL, we developed a corpus that is enhanced by annotation time as a novel form of linguistic meta-data. This corpus allowed for the evaluation of AL in terms of true annotation cost, instead of just the number of examples as an overly simplifying approximation. Experiments showed that our approaches to AL still considerably outperformed random sampling. Additionally, we presented three novel approaches to incorporating annotation costs into AL selection, so that cost and usefulness are considered simultaneously in order to make the best decision about which examples to select. These methods were also applied to make our approach to semi-supervised AL cost-sensitive. Our experiments showed that annotation effort in terms of true annotation time is reduced still further by the cost-sensitive versions of our approaches to AL.

The approaches to the Requirements 6, 7, and 8 were formulated and addressed as cases of multi-criteria AL problems. In Chapter 8, we provided a novel decision-theoretical formulation of AL with multiple selection criteria. The solutions to the above-mentioned three requirements are essentially instantiations of multi-criteria AL in combination with one of the methods of dealing with multiple criteria during selection, as described in that chapter.

14.2 Perspectives

As already stated at the beginning of this thesis, we assume annotation to be the major bottleneck in the application of established methods for NLP to practical scenarios. However, given efficient resource-aware annotation strategies, such as our AL framework for linguistic annotation, the annotation bottleneck might be largely relieved, allowing for a much wider application of such NLP methods – possibly also in areas where their application was previously unthinkable due to the immense overhead resulting from annotation.

AL itself is not new and the potential of AL for reducing annotation effort was claimed by other researchers some time ago (e.g., by Engelson and Dagan (1996)). However, as our survey on practical usage of AL revealed, AL is hardly really applied in practise (Chapter 13). The major reasons for this, as we identified through the

survey, might be summarized by the following two questions: “How can AL for NLP be practically realized without too much overhead in terms of implementation and education of the annotator?” and “Can AL for NLP really cut annotation costs?”.

The first question concerns the public availability of an AL-enabled annotation environment and information about best-practises, user studies and success (or failure) stories – issues best addressed by a community working faced with the same problems. In this spirit, the first workshop on “Active Learning for Natural Language Processing”, held in 2009, aimed at bringing together researchers and users to explore the challenges and opportunities of AL for NLP tasks. Due to the success of this first workshop, a follow-up workshop will be held in 2010.¹ The author of this thesis is one of the co-organizers of both workshops.

The second question indicates that cost-sensitive AL may be the key direction of further research on AL. We presume, that AL may only find broad acceptance as a resource-aware annotation strategy when the reduction of true annotation time can be attested. As part of such research – besides studies on better cost-sensitive sampling strategies – the development of predictive cost models is essential for the actual realization of cost-aware AL. In practise, annotation time is not known prior to annotation and so has to be estimated.

However, we believe that the development of predictive cost models requires a shift of research from machine learning and computational linguistics more towards the field of psycholinguistics. Methods used in research on human-computer interaction, such as tracking of observational data in terms of gaze durations and gaze movements by eye-tracking devices, have already been used successfully in psycholinguistics to understand human language processing (Rayner, 1998). Such methodology might also be used to understand better the cognitive processes behind annotation, based on which accurate models of annotation cost may be developed in the future.

¹A follow-up workshop will be held in conjunction with the 2010 Conference of the North American Chapter of the Association for Computational Linguistics.

Appendix A

Notation

\mathbb{R} : set of real numbers

\mathcal{X} : observation space, set of all possible observations

x : a single observation in \mathcal{X}

\mathcal{Y} : target (label) space, set of all possible target values

y : a specific target in \mathcal{Y}

$l = (x_i, y_i)$: a tuple of observation and its label, also called a *labeled* example

$p = (x_i)$: an unlabeled example consisting of the observation x_i only

\mathcal{L} : set of all labeled examples $l = (x, y)$, $\mathcal{L} \in \mathcal{X} \times \mathcal{Y}$

\mathcal{P} : set of all unlabeled examples $p = (x)$, $\mathcal{P} \in \mathcal{X}$

\mathcal{T} : test set of examples (x, y)

$F(x)$: feature generating function

$f_j(x)$: a feature/feature function

\vec{x} : feature representation of observation x , with $\vec{x} = (f_1(x), \dots, f_k(x))$.

θ : model parameters, with $\theta = (\lambda_1, \dots, \lambda_k)$

T : learning algorithm which induces a model θ from some training data \mathcal{L}

ℓ : objective function for parameter estimation (e.g., log-likelihood function)

$g_\theta(x)$: a classifier/decision function based on model θ

$perf(\theta, \mathcal{T})$: performance of model θ evaluated on test set \mathcal{T}

$u(p, \theta)$: utility scoring function for example p relative to model θ

Appendix B

Abbreviations

AL	Active Learning	p. 29
AUC	Area Under the learning Curve	p. 60
BCR	Benefit-Cost Ratio	p. 189
CCS	Cost-Constrained Sampling	p. 188
CFP	Cost For target Performance	p. 60
CRF	Conditional Random Field	p. 19
CSAL	Cost-Sensitive Active Learning	p. 177
FC	F-Complement	p. 39
FuSAL	Fully-Supervised Active Learning	p. 98
HDM	Hierarchical Decision-Making	p. 144
HLT	Human Language Technology	p. 2
HMM	Hidden Markov Model	p. 16
IAA	Inter-Annotator Agreement	p. 201
IE	Information Extraction	p. 205
i.i.d.	identically and independently distributed	p. 30
ISC	Intrinsic Stopping Criterion	p. 80
KLM	Kullback-Leibler divergence to the Mean	p. 39
LC	Least Confidence	p. 36
Lrk	Linear Rank Combination	p. 189
MADM	Multi-Attribute Decision-Making	p. 142
MAV	Multi-Attribute Value (function)	p. 144
MaxEnt	Maximum Entropy (model)	p. 16
MCAL	Multi-Criteria Active Learning	p. 142
ML	Machine Learning	p. 11
NB	Naïve Bayes	p. 15
NE	Named Entity	p. 50
NER	Named Entity Recognition	p. 49
NLP	Natural Language Processing	p. 1
POS	Part-Of-Speech	p. 51
QbB	Query-by-Bagging	p. 38

QbC	Query-by-Committee	p. 37
RAI	Relative Area Increase	p. 60
RD	Random Sampling	p. 65
SA	Selection Agreement	p. 78
SeSAL	Semi-Supervised Active Learning	p. 98
SVM	Support Vector Model	p. 23
US	Uncertainty Sampling	p. 36
VE	Vote Entropy	p. 38
VSA	Validation Set Agreement	p. 79

Appendix C

Joint Publications

Some of the publications that this thesis is based on were written in cooperation with others than the supervisors of this thesis. The following list describes the respective contributions.

Chapter 2

The introduction to probabilistic models constitutes a condensed version of a technical report published together with Roman Klinger (Klinger and Tomanek, 2007). This publication is based on many discussions between Roman Klinger and the author of this thesis. The author of this thesis mainly focused on the description of the concept of graphical models and the NB, HMM, and MaxEnt models. Roman Klinger described CRFs in detail.

Chapter 4

The NER-specific AL framework is based on joint work with Joachim Wermter (Tomanek et al., 2007a). Joachim Wermter contributed by writing large parts of the introduction and served as a discussion partner during evaluation of the experiments. The author of this thesis has implemented the AL framework, performed the experiments, and written the main part of the paper.

Joint work with Florian Laws and Hinrich Schütze on the co-selection effect was published in (Tomanek et al., 2009). This work was supervised by Udo Hahn and Hinrich Schütze who served as discussion partners and helped during writing. The implementation and experiments are joint work with equally weighted contributions from the author of this thesis and Florian Laws. The author of this thesis implemented sentence-selection AL and performed the respective experiments, whereas Florian Laws focused on token-selection AL.

Chapter 5

The intrinsic stopping criterion is based on joint work with Fredrik Olsson (Olsson and Tomanek, 2009). Fredrik Olsson performed experiments on document-selection AL, whereas the author of this thesis performed experiments on sentence-selection AL and on stream-based AL. Much of the writing of this publication was done by Fredrik Olsson.

Chapter 9

Multi-task AL is based on joint work with Roi Reichart and Ari Rappoport (Reichart et al., 2008). The implementation and experiments are joint work with equally weighted contributions from both the author of this thesis and Roi Reichart. The author of this thesis performed the experiments on the NER task, Roi Reichart focused on the Parsing task. The approaches to multi-task AL were developed in joint discussion. Udo Hahn and Ari Rappoport served as discussion partners and helped during writing.

Chapter 12

The annotation environment for AL-driven annotation is based on joint work with Joachim Wermter (Tomanek et al., 2007b). The annotation environment was developed and implemented by the author of this thesis. Joachim Wermter designed and supervised the real-world annotation projects. Most of the writing of this publication was done by the author of this thesis.

The various annotation endeavours performed between 2006 and 2008 at the computational linguistics unit of the Friedrich-Schiller-University Jena are described in (Hahn et al., 2008). The author of this thesis provided the annotation framework.

Chapter 13

The survey on practical use of AL is based on joint work with Fredrik Olsson (Tomanek and Olsson, 2009). The survey was designed and set-up with equal contributions. The author of this thesis did the analysis and evaluation of the survey results.

Appendix D

Additional Material for Chapter 7

D.1 Correlation Coefficients

Pearson's correlation coefficient is given by:

$$\rho(x, y) = \frac{\text{cov}(x, y)}{\sigma_x \cdot \sigma_y}$$

The weighted correlation coefficient is given by

$$\rho_w(x, y) = \frac{\text{cov}_w(x, y)}{\sqrt{\text{cov}_w(y, y) \cdot \text{cov}_w(y, y)}}$$

with the weighted covariance defined as

$$\text{cov}_w(x, y) = \frac{\sum_{i=1}^n w_i (x_i - \bar{x}_w)(y_i - \bar{y}_w)}{1 - \sum_{i=1}^n w_i^2}$$

where w_i specifies an importance weight for each x_i . The weighted mean \bar{x}_w is defined as

$$\bar{x}_w = \sum_{i=1}^n w_i x_i.$$

Spearman's rank correlation coefficient is calculated by simply translating x and y into ranks $R(x)$ and $R(y)$ and calculating the unweighted or weighted Pearson's correlation coefficient on $R(x)$ and $R(y)$ instead of x and y .

D.2 Weka Parameter Settings

This section gives an overview of the parameter settings used for the WEKA learners used on the UCI data sets.

For J48 applied with Bagging, the class `weka.classifiers.meta.Bagging` was used with these options:

```
-P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
```

For the Naïve Bayes learner, the class `weka.classifiers.bayes.NaiveBayes` was used only with the `-D` option.

For the SVM learner, the class `weka.classifiers.functions.SMO` was used with these options:

```
-C 1.0 -E 1.0 -G 0.01 -A 250007 -L 0.0010  
-P 1.0E-12 -N 0 -M -V -1 -W 1
```

For the MaxEnt learner, the class `weka.classifiers.functions.Logistic` was used with these options:

```
-R 1.0E-8 -M -1
```

D.3 Information on Resampling

Resampling to obtain a NER-like class distribution was done by downsampling the examples of all classes except one. This was only possible for three of the five data sets: The SICK and the MUSHROOM data set did not contain a sufficient number of examples to yield the target distribution. On the SEGMENT, NURSERY and CAR data sets, resampling was done so that all examples of the class, which in the original data set subsumed most examples, were kept and examples for the other classes were removed until a class distribution similar to that of NER was achieved. On the CAR data set, the class “unacc” is the new majority class (with 89 % of all examples), on the NURSERY data set, “not_recomm” is the new majority class (92 %), and on the SEGMENT data set, the class “sky” was chosen as majority class (82 %).

D.4 Additional Data

This section provides results omitted in Chapter 7 for readability. This includes basically detailed RAI and REU scores for the UCI data sets.

CAR data set				
selector	J48	consumer		
		MaxEnt	NB	SVM
J48	0.00	-0.18	-0.43	-0.43
MaxEnt	-0.97	0.00	-1.10	-0.32
NB	-0.88	-0.19	0.00	-0.35
SVM	-0.79	0.37	-0.79	0.00
MUSHROOM data set				
selector	J48	consumer		
		MaxEnt	NB	SVM
J48	0.00	-0.48	-0.73	-0.17
MaxEnt	0.06	0.00	-0.49	0.16
NB	0.07	-0.02	0.00	0.05
SVM	-1.14	-0.53	-1.24	0.00
NURSERY data set				
selector	J48	consumer		
		MaxEnt	NB	SVM
J48	0.00	-1.07	-2.40	-0.85
MaxEnt	-0.21	0.00	-1.61	-0.13
NB	0.66	0.17	0.00	0.08
SVM	-0.12	-0.79	-1.33	0.00
SEGMENT data set				
selector	J48	consumer		
		MaxEnt	NB	SVM
J48	0.00	-0.73	-0.40	-0.88
MaxEnt	-0.78	0.00	-1.23	-3.13
NB	-2.55	-1.41	0.00	-1.59
SVM	-4.07	-1.96	-2.35	0.00
SICK data set				
selector	J48	consumer		
		MaxEnt	NB	SVM
J48	0.00	-0.79	-0.47	-0.83
MaxEnt	0.32	0.00	-0.31	-0.84
NB	0.20	-0.79	0.00	-2.04
SVM	0.36	-0.12	-0.25	0.00

Table D.1: REU scores on UCI data sets for samples of 100 examples.

CAR data set					
selector	J48	selector			unbiased sample
		MaxEnt	NB	SVM	
J48	1.00	0.93	0.92	0.93	0.93
MaxEnt	0.93	1.00	0.92	0.93	0.92
NB	0.92	0.92	1.00	0.92	0.91
SVM	0.93	0.93	0.92	1.00	0.91

MUSHROOM data set					
selector	J48	selector			unbiased sample
		MaxEnt	NB	SVM	
J48	1.00	0.88	0.80	0.80	0.90
MaxEnt	0.88	1.00	0.89	0.86	0.80
NB	0.80	0.89	1.00	0.85	0.67
SVM	0.80	0.86	0.85	1.00	0.71

NURSERY data set					
selector	J48	selector			unbiased sample
		MaxEnt	NB	SVM	
J48	1.00	0.90	0.91	0.91	0.93
MaxEnt	0.90	1.00	0.92	0.92	0.96
NB	0.91	0.92	1.00	0.92	0.94
SVM	0.91	0.92	0.92	1.00	0.93

SEGMENT data set					
selector	J48	selector			unbiased sample
		MaxEnt	NB	SVM	
J48	1.00	0.96	0.94	0.94	0.97
MaxEnt	0.96	1.00	0.93	0.93	0.97
NB	0.94	0.93	1.00	0.94	0.93
SVM	0.94	0.93	0.94	1.00	0.93

SICK data set					
selector	J48	selector			unbiased sample
		MaxEnt	NB	SVM	
J48	1.00	0.94	0.94	0.96	0.94
MaxEnt	0.94	1.00	0.92	0.96	0.96
NB	0.94	0.92	1.00	0.93	0.92
SVM	0.96	0.96	0.93	1.00	0.96

Table D.2: Mutual distributional similarity (SIM score) between the samples of 100 examples selected with AL using different selectors and distributional similarity between these samples and an unbiased sample as represented by the complete pool \mathcal{P} .

CAR data set				
car	J48	consumer		
		MaxEnt	NB	SVM
J48	1.00	0.37	0.22	0.20
MaxEnt	0.17	1.00	0.11	0.48
NB	0.28	0.33	1.00	0.31
SVM	0.08	0.58	0.37	1.00
MUSHROOM data set				
selector	J48	consumer		
		MaxEnt	NB	SVM
J48	1.00	0.32	0.34	0.41
MaxEnt	0.79	1.00	0.11	0.51
NB	0.70	0.40	1.00	0.47
SVM	0.77	0.38	0.14	1.00
NURSERY data set				
selector	J48	consumer		
		MaxEnt	NB	SVM
J48	1.00	0.59	0.63	0.54
MaxEnt	0.55	1.00	0.61	0.66
NB	0.54	0.57	1.00	0.63
SVM	0.50	0.69	0.73	1.00
SEGMENT data set				
selector	J48	consumer		
		MaxEnt	NB	SVM
J48	1.00	0.40	0.54	0.38
MaxEnt	0.20	1.00	0.52	0.15
NB	0.10	0.32	1.00	0.36
SVM	0.18	0.32	0.49	1.00
SICK data set				
selector	J48	consumer		
		MaxEnt	NB	SVM
J48	1.00	0.14	0.24	0.59
MaxEnt	0.28	1.00	0.28	0.64
NB	0.23	0.13	1.00	0.19
SVM	0.31	0.22	0.21	1.00

Table D.3: Similarity of feature ranking (FR score) of a consumer trained on a self-selected sample and a foreign-selected sample. Example on MUSHROOM data set: $FR(J48, MaxEnt)=0.32$ means that the feature ranking of a MaxEnt consumer trained on a sample selected by a MaxEnt selector has a weighted Spearman's rank correlation coefficient with the feature ranking of a MaxEnt consumer trained on a sample selected by a J48 of 0.32.

data set	class distribution
SEGMENT original	(0.14, 0.14, 0.14, 0.14, 0.14, 0.14, 0.14)
SEGMENT resampled	(0.82 , 0.06, 0.06, 0.06, 0.00, 0.00, 0.00)
NURSERY original	(0.33, 0.33, 0.31, 0.03)
NURSERY resampled	(0.92 , 0.04, 0.04, 0.00)
CAR original	(0.70, 0.22, 0.04, 0.04)
CAR resampled	(0.89 , 0.04, 0.04, 0.04)

Table D.4: Class distribution in original and resampled UCI data sets. Highlighted number refers to percentage of examples of the majority class in the resampled data sets.

CAR resampled data set					
selector	REU score according to consumers				RAI score
	J48	MaxEnt	NB	SVM	
J48	0.00	−0.93	−0.89	−0.90	0.43
MaxEnt	−0.47	0.00	−0.73	−0.25	6.51
NB	−2.56	0.15	0.00	−0.00	3.74
SVM	−4.31	0.15	−0.67	0.00	9.45

NURSERY resampled data set					
selector	REU score according to consumers				RAI score
	J48	MaxEnt	NB	SVM	
J48	0.00	−3.70	−0.86	−0.27	1.50
MaxEnt	−0.06	0.00	−0.29	−0.11	1.37
NB	0.45	−3.38	0.00	0.13	3.02
SVM	0.23	−4.25	−0.79	0.00	2.09

SEGMENT resampled data set					
selector	REU score according to consumers				RAI score
	J48	MaxEnt	NB	SVM	
J48	0.00	0.51	−0.09	0.07	3.74
MaxEnt	−0.26	0.00	−0.56	−0.55	1.74
NB	0.01	0.59	0.00	0.06	1.34
SVM	−0.11	0.36	−0.29	0.00	1.18

Table D.5: REU and RAI scores for resampled data sets.

Bibliography

- [Abe and Mamitsuka 1998] ABE, Naoki; MAMITSUKA, Hiroshi: Query Learning Strategies Using Boosting and Bagging. In: *ICML'98: Proceedings of the 15th International Conference on Machine Learning*, Morgan Kaufmann, 1998, pp. 1–9.
- [Airoldi et al. 2006] AIROLDI, Edoardo; LESKOVEC, Jure; KLEINBERG, Jon; TENENBAUM, Josh (Editors.): *NIPS 2006 Workshop on Learning when test and training inputs have different distributions*. 2006.
- [Alias-i 2008] ALIAS-I: *LingPipe 3.7.1*. 2008. – <http://alias-i.com/lingpipe/>, date accessed 01/01/2008.
- [Anderson and Moore 2005] ANDERSON, Brigham; MOORE, Andrew: Active learning for Hidden Markov Models: objective functions and algorithms. In: *ICML'05: Proceedings of the 22nd international conference on Machine learning*, ACM, 2005, pp. 9–16.
- [Angluin 1988] ANGLUIN, Dana: Queries and Concept Learning. In: *Machine Learning* 2 (1988), No. 4, pp. 319–342.
- [Arora et al. 2009] ARORA, Shilpa; NYBERG, Eric; ROSÉ, Carolyn: Estimating Annotation Cost for Active Learning in a Multi-Annotator Environment. In: *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, Association for Computational Linguistics, 2009, pp. 18–26.
- [Asuncion and Newman 2007] ASUNCION, Arthur; NEWMAN, David: *UCI Machine Learning Repository*. 2007. – <http://archive.ics.uci.edu/ml/>, date accessed 04/01/2009.
- [Atkinson and Donev 1992] ATKINSON, Anthony C.; DONEV, Alexander: *Optimum experimental designs*. Oxford Science Publications, 1992.
- [Baldrige and Osborne 2004] BALDRIDGE, Jason; OSBORNE, Miles: Active Learning and the Total Cost of Annotation. In: *EMNLP'04: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2004, pp. 9–16.

- [Becker and Osborne 2005] BECKER, Markus; OSBORNE, Miles: A Two-Stage Method for Active Learning of Statistical Grammars. In: *IJCAI'05: Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Professional Book Center, 2005, pp. 991–996.
- [Benajiba et al. 2008] BENAJIBA, Yassine; DIAB, Mona; ROSSO, Paolo: Arabic Named Entity Recognition using Optimized Feature Sets. In: *EMNLP'08: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2008, pp. 284–293.
- [Berger et al. 1996] BERGER, Adam; DELLA PIETRA, Stephen; DELLA PIETRA, Vincent: A Maximum Entropy Approach to Natural Language Processing. In: *Computational Linguistics* 22 (1996), No. 1, pp. 39–71.
- [Bickel 2005] BICKEL, Daniel: *Code developed at the University of Pennsylvania*. 2005. – <http://www.cis.upenn.edu/dbikel/software.html>, date accessed 01/01/2008.
- [Bickel et al. 2007] BICKEL, Steffen; BRÜCKNER, Michael; SCHEFFER, Tobias: Discriminative learning for differing training and test distributions. In: *ICML'07: Proceedings of the 24th International Conference on Machine learning*, ACM, 2007, pp. 81–88.
- [Bikel et al. 1997] BIKEL, Daniel; MILLER, Scott; SCHWARTZ, Richard; WEISCHEDEL, Ralph: Nymble: A high-performance learning name-finder. In: *ANLP'97: Proceedings of the 5th conference on Applied Natural Language Processing*, Association for Computational Linguistics, 1997, pp. 194–201.
- [Bishop 2006] BISHOP, Christopher: *Pattern Recognition and Machine Learning*. Springer Berlin/Heidelberg, 2006.
- [Black et al. 1998] BLACK, William; RINALDI, Fabio; MOWATT, David: Facile: Description Of The NE System Used For MUC-7. In: *MUC-7: Proceedings of the 7th Message Understanding Conference*, 1998. – available on-line: <http://acl.ldc.upenn.edu/muc7/>.
- [Bloodgood and Shanker 2009a] BLOODGOOD, Michael; SHANKER, Vijay: A Method for Stopping Active Learning Based on Stabilizing Predictions and the Need for User-Adjustable Stopping. In: *CoNLL'09: Proceedings of the 13th Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2009, pp. 39–47.
- [Bloodgood and Shanker 2009b] BLOODGOOD, Michael; SHANKER, Vijay: Taking into Account the Differences between Actively and Passively Acquired Data: The

- Case of Active Learning with Support Vector Machines for Imbalanced Datasets. In: *NAACL'09: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 2009, pp. 137–140.
- [Blum and Mitchell 1998] BLUM, Avrim; MITCHELL, Tom: Combining labeled and unlabeled data with co-training. In: *COLT'98: Proceedings of the 11th Annual Conference on Computational Learning Theory*, ACM, 1998, pp. 92–100.
- [Borthwick et al. 1998] BORTHWICK, Andrew; STERLING, John; AGICHTEIN, Eugene; GRISHMAN, Ralph: NYU: Description of the MENE Named Entity System as Used in MUC-7. In: *MUC-7: Proceedings of the 7th Message Understanding Conference*, 1998. – available on-line: <http://acl1.ldc.upenn.edu/muc7/>.
- [Breiman 1996] BREIMAN, Leo: Bagging predictors. In: *Machine Learning* 24 (1996), No. 2, pp. 123–140.
- [Brinker 2003] BRINKER, Klaus: Incorporating diversity in active learning with support vector machines. In: *ICML'03: Proceedings of the 20th International Conference on Machine Learning*, AAAI, 2003, pp. 59–66.
- [Buyko et al. 2009] BUYKO, Ekaterina; FAESSLER, Erik; WERMTER, Joachim; HAHN, Udo: Event Extraction from Trimmed Dependency Graphs. In: *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, Association for Computational Linguistics, 2009, pp. 19–27.
- [Caruana 1997] CARUANA, Rich: Multitask Learning. In: *Machine Learning* 28 (1997), No. 1, pp. 41–75.
- [Chapelle et al. 2006] CHAPELLE, Olivier; SCHÖLKOPF, Bernhard; ZIEN, Alexander (Editors.): *Semi-Supervised Learning*. MIT Press, 2006.
- [Chawla et al. 2002] CHAWLA, Nitesh; BOWYER, Kevin; HALL, Lawrence; KEGELMEYER, Philip: SMOTE: Synthetic Minority Over-sampling Technique. In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357.
- [Cheng et al. 2008] CHENG, Haibin; ZHANG, Ruofei; PENG, Yefei; MAO, Jianchang; TAN, Pang-Ning: Maximum Margin Active Learning for Sequence Labeling with Different Length. In: *ICDM'08: Proceedings of the 8th Industrial Conference on Advances in Data Mining*, Springer Berlin/Heidelberg, 2008, pp. 345–359.
- [Claire et al. 2005] CLAIRE, Grover; BECKER, Markus; HACHEY, Ben; ALEX, Beatrice: Optimising Selective Sampling for Bootstrapping Named Entity Recognition. In: *Proceedings of the ICML-2005 Workshop on Learning with Multiple Views*, ACM, 2005, pp. 5–11.

- [Cohen 1960] COHEN, Jacob: A Coefficient of Agreement for Nominal Scales. In: *Educational and Psychological Measurement* 20 (1960), No. 1, pp. 37–46.
- [Cohn et al. 1994] COHN, David; ATLAS, Les; LADNER, Richard: Improving Generalization with Active Learning. In: *Machine Learning* 15 (1994), No. 2, pp. 201–221.
- [Cohn et al. 1996] COHN, David; GHAHRAMANI, Zoubin; JORDAN, Michael: Active Learning with Statistical Models. In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 129–145.
- [Collins 2003] COLLINS, Michael: Head-Driven Statistical Models for Natural Language Parsing. In: *Computational Linguistics* 29 (2003), No. 4, pp. 589–637.
- [Cortes et al. 2008] CORTES, Corinna; MOHRI, Mehryar; RILEY, Michael; ROSTAMIZADEH, Afshin: Sample Selection Bias Correction Theory. In: *Algorithmic Learning Theory*, Springer, 2008, pp. 38–52.
- [Dagan and Engelson 1995] DAGAN, Ido; ENGELSON, Sean: Committee-Based Sampling For Training Probabilistic Classifiers. In: *ICML'95: Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, 1995, pp. 150–157.
- [Danziger et al. 2007] DANZIGER, Samuel; ZENG, Jue; WANG, Ying; BRACHMANN, Rainer; LATHROP, Richard: Choosing where to look next in a mutation sequence space: Active Learning of informative p53 cancer rescue mutants. In: *Bioinformatics* 23 (2007), No. 13, pp. 104–114.
- [Darroch and Ratcliff 1972] DARROCH, J. N.; RATCLIFF, D.: Generalized Iterative Scaling for Log-Linear Models. In: *The Annals of Mathematical Statistics* 43 (1972), No. 5, pp. 1470–1480.
- [Dasgupta 2004] DASGUPTA, Sanjoy: Analysis of a greedy active learning strategy. In: *Advances in Neural Information Processing Systems*, MIT Press, 2004, pp. 337–344.
- [Dasgupta 2006] DASGUPTA, Sanjoy: Coarse sample complexity bounds for active learning. In: *Advances in Neural Information Processing Systems*, MIT Press, 2006, pp. 235–242.
- [Dempster et al. 1977] DEMPSTER, Arthur; LAIRD, Nan; RUBIN, Donald: Maximum likelihood from incomplete data via the EM algorithm. In: *Journal of the Royal Statistical Society* 39 (1977), No. 1, pp. 1–38.

- [Donmez and Carbonell 2008] DONMEZ, Pinar; CARBONELL, Jaime: Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In: *CIKM'08: Proceeding of the 17th ACM conference on Information and knowledge management*, ACM, 2008, pp. 619–628.
- [Dwyer and Holte 2007] DWYER, Kenneth; HOLTE, Robert: Decision Tree Instability and Active Learning. In: *ECML'07: Proceedings of 18th European Conference on Machine Learning*, Springer, 2007, pp. 128–139.
- [Ehrgott 2000] EHRGOTT, Matthias: *Multicriteria optimization*. Springer Berlin/Heidelberg, 2000.
- [Elkan 2001] ELKAN, Charles: The Foundations of Cost-Sensitive Learning. In: *IJCAI'01: Proceedings of the 17th International Joint Conference of Artificial Intelligence*, Morgan Kaufmann, 2001, pp. 973–978.
- [Engelson and Dagan 1996] ENGELSON, Sean; DAGAN, Ido: Minimizing manual annotation cost in supervised training from corpora. In: *ACL'96: Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 1996, pp. 319–326.
- [Ertekin et al. 2007] ERTEKIN, Seyda; HUANG, Jian; BOTTOU, Leon; GILES, Lee: Learning on the border: active learning in imbalanced data classification. In: *CIKM'07: Proceedings of the 16th ACM conference on Conference on information and knowledge management*, ACM, 2007, pp. 127–136.
- [Everitt et al. 2001] EVERITT, Brian; LANDAU, Sabine; LEESE, Morven: *Cluster Analysis*. 4th. Wiley, 2001.
- [Fan et al. 2008] FAN, Rong-En; CHANG, Kai-Wei; HSIEH, Cho-Jui; WANG, Xiang-Rui; LIN, Chih-Jen: LIBLINEAR: A Library for Large Linear Classification. In: *Journal of Machine Learning Research* 9 (2008), pp. 1871–1874.
- [Fan et al. 2005] FAN, Wei; DAVIDSON, Ian; ZADROZNY, Bianca; YU, Philip: An Improved Categorization of Classifier's Sensitivity on Sample Selection Bias. In: *ICDM'05: Proceedings of the 5th IEEE International Conference on Data Mining*, IEEE Computer Society, 2005, pp. 605–608.
- [Finkel et al. 2005] FINKEL, Jenny R.; GRENAGER, Trond; MANNING, Christopher: Incorporating non-local information into information extraction systems by Gibbs sampling. In: *ACL'05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2005, pp. 363–370.

- [Finkel and Manning 2009] FINKEL, Jenny R.; MANNING, Christopher: Joint Parsing and Named Entity Recognition. In: *HLT'09: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 2009, pp. 326–334.
- [Freund and Schapire 1996] FREUND, Yoav; SCHAPIRE, Robert: Experiments with a New Boosting Algorithm. In: *ICML'96: Proceedings of the 13th International Conference on Machine Learning*, Morgan Kaufmann, 1996, pp. 148–156.
- [Freund et al. 1997] FREUND, Yoav; SEUNG, Sebastian; SHAMIR, Eli; TISHBY, Naftali: Selective Sampling Using the Query by Committee Algorithm. In: *Machine Learning* 28 (1997), No. 2-3, pp. 133–168.
- [Fukuda et al. 1998] FUKUDA, Ken-Ichiro; TSUNODA, T.; TAMURA, A.; TAKAGI, Toshihisa: Toward information extraction: Identifying protein names from biological papers. In: *PSB'98: Proceedings of the 3rd Pacific Symposium on Biocomputing*, World Scientific Publishing, 1998, pp. 705–716.
- [Geman et al. 1992] GEMAN, Stuart; BIENENSTOCK, Elie; DOURSAT, René: Neural Networks and the Bias/Variance Dilemma. In: *Neural Computation* 4 (1992), No. 1, pp. 1–58.
- [Getoor and Taskar 2007] GETOOR, Lise; TASKAR, Ben (Editors.): *Introduction to Statistical Relational Learning*. MIT Press, 2007.
- [Gilad-Bachrach et al. 2006] GILAD-BACHRACH, Ran; NAVOT, Amir; TISHBY, Naftali: Query by Committee Made Real. In: *NIPS'05: Advances in Neural Information Processing Systems*, MIT Press, 2006, pp. 443–450.
- [Grishman and Sundheim 1996] GRISHMAN, Ralph; SUNDHEIM, Beth: Message Understanding Conference–6: A brief history. In: *COLING'96: Proceedings of the 16th International Conference on Computational Linguistics*, Association for Computational Linguistics, 1996, pp. 466–471.
- [Guo and Greiner 2007] GUO, Yuhong; GREINER, Russell: Optimistic Active-Learning Using Mutual Information. In: *IJCAI'07: International Joint Conference on Artificial Intelligence*, Springer, 2007, pp. 823–829.
- [Hachey et al. 2005] HACHEY, Ben; ALEX, Beatrice; BECKER, Markus: Investigating the Effects of Selective Sampling on the Annotation Task. In: *CoNLL'05: Proceedings of the 9th Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2005, pp. 144–151.

- [Haertel et al. 2008a] HAERTEL, Robbie; RINGGER, Eric; SEPPI, Kevin; CARROLL, James; PETER, McClanahan: Assessing the Costs of Sampling Methods in Active Learning for Annotation. In: *Proceedings of ACL-08: HLT, Short Papers*, Association for Computational Linguistics, 2008, pp. 65–68.
- [Haertel et al. 2008b] HAERTEL, Robbie; SEPPI, Kevin; RINGGER, Eric; CARROLL, James: Return on Investment for Active Learning. In: *Proceedings of the NIPS 2008 Workshop on Cost-Sensitive Machine Learning*, MIT Press, 2008.
- [Haffari et al. 2009] HAFFARI, Gholamreza; ROY, Maxim; SARKAR, Anoop: Active Learning for Statistical Phrase-based Machine Translation. In: *HLT'09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 2009, pp. 415–423.
- [Hahn et al. 2008] HAHN, Udo; BEISSWANGER, Elena; BUYKO, Ekaterina; POPRAT, Michael; TOMANEK, Katrin; WERMTER, Joachim: Semantic Annotations for Biology: A Corpus Development Initiative at the Jena University Language & Information Engineering (JULIE) Lab. In: *LREC'08: Proceedings of the 6th International Conference on Language Resources and Evaluation*, European Language Resources Association, 2008, pp. 2257–2261.
- [Hansen and Heskes 2000] HANSEN, Jakob; HESKES, Tom: General Bias/Variance Decomposition with Target Independent Variance of Error Functions Derived from the Exponential Family of Distributions. In: *ICPR'00: Proceedings of 15th International Conference on Pattern Recognition*, IEEE, 2000, pp. 207–210.
- [Heckman 1979] HECKMAN, James: Sample Selection Bias as a Specification Error. In: *Econometrica* 47 (1979), No. 1, pp. 153–61.
- [Hirschman et al. 2005] HIRSCHMAN, Lynette; YEH, Alexander; BLASCHKE, Christian; VALENCIA, Alfonso: Overview of BIOCREATiVE: Critical assessment of information extraction for biology. In: *BMC Bioinformatics* 6 (2005), pp. S1.
- [Ho et al. 1994] HO, Tin K.; HULL, Jonathan; SRIHARI, Sargur: Decision Combination in Multiple Classifier Systems. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16 (1994), No. 1, pp. 66–75.
- [Hoi et al. 2006] HOI, Steven; JIN, Rong; ZHU, Jianke; LYU, Michael: Batch mode active learning and its application to medical image classification. In: *ICML'06: Proceedings of the 23rd international conference on Machine learning*, ACM, 2006, pp. 417–424.

- [Hsu and Taksa 2005] HSU, Frank; TAKSA, Isak: Comparing Rank and Score Combination Methods for Data Fusion in Information Retrieval. In: *Information Retrieval* 8 (2005), No. 3, pp. 449–480.
- [Huang et al. 2009] HUANG, Fang-Lan; HSIEH, Cho-Jui; CHANG, Kai-Wei; LIN, Chih-Jen: Iterative Scaling and Coordinate Descent Methods for Maximum Entropy. In: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, Association for Computational Linguistics, 2009, pp. 285–288.
- [Huang et al. 2008] HUANG, T. S.; DAGLI, C. K.; RAJARAM, S.; CHANG, E. Y.; MANDEL, M. I.; POLINER, G. E.; ELLIS, D. P. W.: Active Learning for Interactive Multimedia Retrieval. In: *Proceedings of the IEEE* 96 (2008), No. 4, pp. 648–667.
- [Hwa 2000] HWA, Rebecca: Sample selection for statistical grammar induction. In: *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in Natural Language Processing and Very Large Corpora*, Association for Computational Linguistics, 2000, pp. 45–52.
- [Hwa 2001] HWA, Rebecca: On minimizing training corpus for parser acquisition. In: *ConLL'01: Proceedings of the 2001 workshop on Computational Natural Language Learning*, Association for Computational Linguistics, 2001, pp. 1–6.
- [Iria 2009] IRIA, José: Automating knowledge capture in the aerospace domain. In: *K-CAP'09: Proceedings of the 5th international conference on Knowledge Capture*, ACM, 2009, pp. 97–104.
- [Isozaki and Kazawa 2002] ISOZAKI, Hideki; KAZAWA, Hideto: Efficient support vector classifiers for named entity recognition. In: *Proceedings of the 19th international conference on Computational linguistics*, Association for Computational Linguistics, 2002, pp. 1–7.
- [Japkowicz 2000] JAPKOWICZ, Nathalie: The Class Imbalance Problem: Significance and Strategies. In: *IC-AI'00: Proceedings of the 2000 International Conference on Artificial Intelligence*, CSREA Press, 2000, pp. 111–117.
- [Jaynes 1957] JAYNES, Edwin: Information Theory and Statistical Mechanics. In: *Physical Review* 106 (1957), No. 4, pp. 620–630.
- [John and Langley 1996] JOHN, George; LANGLEY, Pat: Static Versus Dynamic Sampling for Data Mining. In: *KDD'96: Proceedings of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining*, AAAI, 1996, pp. 367–370.

- [Jurafsky and Martin 2000] JURAFSKY, Daniel; MARTIN, James: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2000.
- [Kääriäinen 2006] KÄÄRIÄINEN, Matti: Active Learning in the Non-realizable Case. In: *ALT'06: Proceedings of the 17th International Conference on Algorithmic Learning Theory*, Springer Berlin/Heidelberg, 2006, pp. 63–77.
- [Kapoor et al. 2007] KAPOOR, Ashish; HORVITZ, Eric; BASU, Sumit: Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning. In: *IJCAI'07: Proceedings of the 20th international joint conference on Artificial intelligence*, Morgan Kaufmann, 2007, pp. 877–882.
- [Keeney and Raiffa 1976] KEENEY, Ralph; RAIFFA, Howard: *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, 1976.
- [Kim et al. 2004] KIM, Jin-Dong; OHTA, Tomoko; TSURUOKA, Yoshimasa; TATEISI, Yuka; COLLIER, Nigel: Introduction to the bio-entity recognition task at JNLPBA. In: *JNLPBA'04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, Association for Computational Linguistics, 2004, pp. 70–75.
- [Kim and Tsujii 2006] KIM, Jin-Dong; TSUJII, Jun'ichi: Corpora and their annotation. In: ANANIADOU, Sophia; MCNAUGHT, John (Editors.): *Text Mining for Biology and Biomedicine*. Artech House, 2006, Chap. 8, pp. 179–211.
- [Kim et al. 2006] KIM, Seokhwan; SONG, Yu; KIM, Kyungduk; CHA, Jeongwon; LEE, Gary G.: MMR-based Active Machine Learning for Bio Named Entity Recognition. In: *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, Association for Computational Linguistics, 2006, pp. 69–72.
- [Klinger et al. 2007] KLINGER, Roman; FRIEDRICH, Christoph; FLUCK, Julianne; HOFMANN-APITIUS, Martin: Named Entity Recognition with Combinations of Conditional Random Fields. In: *Proceedings of the 2nd BioCreative Challenge Evaluation Workshop*, 2007, pp. 89–91.
- [Klinger et al. 2008] KLINGER, Roman; KOLÁŘIK, Corinna; FLUCK, Julianne; HOFMANN-APITIUS, Martin; FRIEDRICH, Christoph: Detection of IUPAC and IUPAC-like Chemical Names. In: *Bioinformatics* 24 (2008), No. 13, pp. 268–276.
- [Klinger and Tomanek 2007] KLINGER, Roman; TOMANEK, Katrin: Classical Probabilistic Models and Conditional Random Fields / Department of Computer Science, Dortmund University of Technology. 2007 (TR07-2-013). – Algorithm Engineering Report.

- [Kohavi and John 1997] KOHAVI, Ron; JOHN, George: Wrappers for feature subset selection. In: *Artificial Intelligence 97* (1997), No. 1-2, pp. 273–324.
- [Körner and Wrobel 2006] KÖRNER, Christine; WROBEL, Stefan: Multi-class Ensemble-Based Active Learning. In: *ECML'06: Proceedings of the 17th European Conference on Machine Learning*, Springer, 2006, pp. 687–694.
- [Kristjansson et al. 2004] KRISTJANSSON, Trausti; CULOTTA, Aron; VIOLA, Paul: Interactive information extraction with constrained Conditional Random Fields. In: *AAAI'04: Proceedings of 19th National Conference on Artificial Intelligence*, AAAI, 2004, pp. 412–418.
- [Krogh and Vedelsby 1995] KROGH, Anders; VEDELSBY, Jesper: Neural Network Ensembles, Cross Validation, and Active Learning. In: *NIPS'95: Advances in Neural Information Processing Systems*, MIT Press, 1995, pp. 231–238.
- [Kulick et al. 2004] KULICK, Seth; BIES, Ann; LIBERMAN, Mark; MANDEL, Mark; McDONALD, Ryan; PALMER, Martha; SCHEIN, Andrew I.: Integrated annotation for biomedical information extraction. In: *Proceedings of the HLT-NAACL 2004 Workshop 'Linking Biological Literature, Ontologies and Databases: Tools for Users'*, Association for Computational Linguistics, 2004, pp. 61–68.
- [Kullback and Leibler 1951] KULLBACK, Solomon; LEIBLER, Richard: On Information and Sufficiency. In: *The Annals of Mathematical Statistics* 22 (1951), No. 1, pp. 79–86.
- [Kuncheva and Whitaker 2003] KUNCHEVA, Ludmila; WHITAKER, Christopher: Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. In: *Machine Learning* 51 (2003), No. 2, pp. 181–207.
- [Lafferty et al. 2001] LAFFERTY, John; MCCALLUM, Andrew; PEREIRA, Fernando: Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In: *ICML'01: Proceedings of the 18th International Conference on Machine Learning*, Morgan Kaufmann, 2001, pp. 282–289.
- [Laws and Schütze 2008] LAWS, Florian; SCHÜTZE, Hinrich: Stopping criteria for active learning of named entity recognition. In: *COLING'08: Proceedings of the 22th International Conference on Computational Linguistics*, Coling 2008 Organizing Committee, 2008, pp. 465–472.
- [Lewis and Catlett 1994] LEWIS, David; CATLETT, Jason: Heterogeneous uncertainty sampling for supervised learning. In: *ICML'94: Proceedings of 11th International Conference on Machine Learning*, Morgan Kaufmann, 1994, pp. 148–156.

- [Lewis and Gale 1994] LEWIS, David; GALE, William: A sequential algorithm for training text classifiers. In: *SIGIR'94 – Proceedings of the 17th annual International Conference on Research and Development in Information Retrieval*, Springer New York, 1994, pp. 3–12.
- [Lichtenwald 2009] LICHTENWALD, Oleg: *Einsatz von Active-Learning-Methoden bei der Erkennung von Named Entities in Nominalphrasen*, Friedrich-Schiller-Universität Jena, Master Thesis, 2009.
- [Lin 1991] LIN, Jianhua: Divergence measures based on the Shannon entropy. In: *IEEE Transactions on Information theory* 37 (1991), No. 1, pp. 145–151.
- [Ling and Du 2008] LING, Charles; DU, Jun: Active learning with direct query construction. In: *KDD'08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2008, pp. 480–487.
- [Linguistic Data Consortium 2001] LINGUISTIC DATA CONSORTIUM: *Message Understanding Conference (MUC) 7*. LDC2001T02. FTP FILE. 2001. – Philadelphia: Linguistic Data Consortium.
- [Liu et al. 2008] LIU, Alexander; JUN, Goo; GHOSH, Joydeep: Active Learning with Spatially Sensitive Labeling Costs. In: *Proceedings of the NIPS 2008 Workshop on Cost-Sensitive Machine Learning*, MIT Press, 2008.
- [Liu 2004] LIU, Ying: Active learning with support vector machine applied to gene expression data for cancer classification. In: *Journal of Chemical Information and Modeling* 44 (2004), pp. 1936–1941.
- [Marcus et al. 1993] MARCUS, Mitchell; SANTORINI, Beatrice; MARCINKIEWICZ, Mary A.: Building a large annotated corpus of English: The Penn Treebank. In: *Computational Linguistics* 19 (1993), No. 2, pp. 313–330.
- [Margineantu 2005] MARGINEANTU, Dragos: Active Cost-Sensitive Learning. In: *IJCAI'05: Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, 2005, pp. 1622–1623.
- [Marsh and Perzanowski 1998] MARSH, Elaine; PERZANOWSKI, Dennis: MUC-7 evaluation of IE technology: Overview of results. In: *MUC-7: Proceedings of the Seventh Message Understanding Conference*, 1998.
- [McCallum 2002] MCCALLUM, Andrew: Mallet: A Machine Learning for Language Toolkit. 2002. – <http://mallet.cs.umass.edu>.

- [McCallum 2009] MCCALLUM, Andrew: Joint inference for natural language processing. In: *CoNLL'09: Proceedings of the 13th Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2009, pp. 1–1.
- [McCallum and Li 2003] MCCALLUM, Andrew; LI, Wei: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: *HLT-NAACL'03: Proceedings of the 7th conference on Natural language learning*, Association for Computational Linguistics, 2003, pp. 188–191.
- [McCallum and Nigam 1998] MCCALLUM, Andrew; NIGAM, Kamal: Employing EM and pool-based Active Learning for text classification. In: *ICML'98: Proceedings of the 15th International Conference on Machine Learning*, AAAI, 1998, pp. 350–358.
- [McDonald and Pereira 2005] McDONALD, Ryan; PEREIRA, Fernando: Identifying Gene and Protein Mentions in Text Using Conditional Random Fields. In: *BMC Bioinformatics* 6 (2005).
- [McDonald et al. 2004] McDONALD, Ryan; WINTERS, Scott; MANDEL, Mark; JIN, Yang; WHITE, Peter; PEREIRA, Fernando: An entity tagger for recognizing acquired genomic variations in cancer literature. In: *Bioinformatics* 20 (2004), No. 17, pp. 3249–3251.
- [Melville and Mooney 2003] MELVILLE, Prem; MOONEY, Raymond: Constructing diverse classifier ensembles using artificial training examples. In: *IJCAI'03: Proceedings of 18th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, 2003, pp. 505–510.
- [Melville and Mooney 2004] MELVILLE, Prem; MOONEY, Raymond: Diverse ensembles for active learning. In: *ICML'04: Proceedings of the 21st International Conference on Machine Learning*, ACM, 2004, pp. 584–59.
- [Melville et al. 2005] MELVILLE, Prem; SAAR-TSECHANSKY, Maytal; PROVOST, Foster; MOONEY, Raymond: An Expected Utility Approach to Active Feature-Value Acquisition. In: *ICDM'05: Proceedings of the 5th IEEE International Conference on Data Mining*, IEEE Computer Society, 2005, pp. 745–748.
- [Mitchell 1997] MITCHELL, Tom: *Machine Learning*. McGraw Hill, 1997.
- [Morales et al. 2004] MORALES, Antonio; CHINELLATO, Eris; FAGG, Andrew; POBIL, Angel del: An active learning approach for assessing robot grasp reliability. In: *IROS'04: Proceedings on International Conference on Intelligent Robots and Systems*, IEEE, 2004, pp. 485–490.

- [Muslea et al. 2000] MUSLEA, Ion A.; MINTON, Steven; KNOBLOCK, Craig: Selective Sampling with Redundant Views. In: *Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, AAAI, 2000, pp. 621–626.
- [Muslea et al. 2002] MUSLEA, Ion A.; MINTON, Steven; KNOBLOCK, Craig: Active semi-supervised learning = Robust multi-view learning. In: *ICML'02: Proceedings of the 19th International Conference on Machine Learning*, Morgan Kaufmann, 2002, pp. 435–442.
- [Müller and Strube 2003] MÜLLER, Christoph; STRUBE, Michael: Multi-level annotation in MMAX. In: *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*, Association for Computational Linguistics, 2003, pp. 198–207.
- [Nadeau and Sekine 2007] NADEAU, David; SEKINE, Satoshi: A survey of named entity recognition and classification. In: *Linguisticae Investigationes* 30 (2007), No. 1, pp. 3–26.
- [Ngai and Yarowsky 2000] NGAI, Grace; YAROWSKY, David: Rule writing or annotation: cost-efficient resource usage for base noun phrase chunking. In: *ACL'00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2000, pp. 117–125.
- [Ninio 2006] NINIO, Anat: *Language and the Learning Curve*. Oxford University Press, 2006.
- [Olsson 2008] OLSSON, Fredrik: *Bootstrapping Named Entity Annotation by Means of Active Machine Learning: A Method for Creating Corpora*, University of Gothenburg, Dissertation, 2008.
- [Olsson 2009] OLSSON, Fredrik: A literature survey of active machine learning in the context of natural language processing / Swedish Institute of Computer Science. 2009 (T2009:06). – SICS Technical Report.
- [Olsson and Tomanek 2009] OLSSON, Fredrik; TOMANEK, Katrin: An Intrinsic Stopping Criterion for Committee-Based Active Learning. In: *CoNLL'09: Proceedings of the 13th Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2009, pp. 138–146.
- [OpenNLP 2008] OPENNLP: *OpenNLP 1.3*. 2008. – <http://opennlp.sourceforge.net/>, date accessed 01/01/2008.
- [Palmer et al. 2009] PALMER, Alexis; MOON, Taesun; BALDRIDGE, Jason: Evaluating automation strategies in language documentation. In: *Proceedings of the*

- NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, Association for Computational Linguistics, 2009, pp. 36–44.
- [Pierce and Cardie 2001] PIERCE, David; CARDIE, Claire: Limitations of co-training for natural language learning from large datasets. In: *EMNLP'01: Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2001, pp. 1–9.
- [Provost 2000] PROVOST, Foster: Machine Learning from Imbalanced Data Sets 101 (Extended Abstract). In: *Learning from Imbalanced Data Sets: Papers from the AAAI Workshop*, AAAI, 2000, pp. 1–3.
- [Provost et al. 1999] PROVOST, Foster; JENSEN, David; OATES, Tim: Efficient progressive sampling. In: *KDD'99: Proceedings of the 5th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, ACM, 1999, pp. 23–32.
- [Pustejovsky et al. 2003] PUSTEJOVSKY, James; HANKS, Patrick; SAURÍ, Roser; SEE, Andrew; GAIZAUSKAS, Robert; SETZER, Andrea; RADEV, Dragomir; SUNDEHEIM, Beth; DAY, David; FERRO, Lisa; LAZO, Marcia: The TimeBank Corpus. In: *Proceedings of the Corpus Linguistics 2003 Conference*, Lancaster University, 2003, pp. 647–656.
- [Quinlan 1993] QUINLAN, Ross: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [Rabiner 1989] RABINER, Lawrence: A tutorial on Hidden Markov Models and selected applications in speech recognition. In: *Proceedings of IEEE* 77 (1989), No. 2, pp. 257–286.
- [Ratnaparkhi 1999] RATNAPARKHI, Adwait: Learning to Parse Natural Language with Maximum Entropy Models. In: *Machine Learning* 34 (1999), No. 1-3, pp. 151–175.
- [Rayner 1998] RAYNER, Keith: Eye movements in reading and information processing: 20 years of research. In: *Psychological Bulletin* 124 (1998), November, No. 3, pp. 372–422.
- [Reichart and Rappoport 2007] REICHART, Roi; RAPPOPORT, Ari: An Ensemble Method for Selection of High Quality Parses. In: *ACL'07: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2007, pp. 408–415.

- [Reichart et al. 2008] REICHART, Roi; TOMANEK, Katrin; HAHN, Udo; RAPPOPORT, Ari: Multi-Task Active Learning for Linguistic Annotations. In: *ACL'08: Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics*, Association for Computational Linguistics, 2008, pp. 861–869.
- [Riccardi and Hakkani-Tür 2005] RICCARDI, Giuseppe; HAKKANI-TÜR, Dilek: Active Learning for Automatic Speech Recognition. In: *IEEE Transactions on Speech and Audio Processing*, Institute of Electrical and Electronics Engineers, 2005, pp. 504–511.
- [van Rijsbergen 1979] RIJSBERGEN, Cornelis van: *Information Retrieval*. 2nd. 1979.
- [Ringger et al. 2008] RINGGER, Eric; CARMEN, Marc; HAERTEL, Robbie; SEPPI, Kevin; LONSDALE, Deryle; MCCLANAHAN, Peter; CARROLL, James; ELLISON, Noel: Assessing the Costs of Machine-Assisted Corpus Annotation through a User Study. In: *LREC'08: Proceedings of the 6th International Language Resources and Evaluation*, European Language Resources Association, 2008, pp. 3318–3324.
- [Ringger et al. 2007] RINGGER, Eric; MCCLANAHAN, Peter; HAERTEL, Robbie; BUSBY, George; CARMEN, Marc; CARROLL, James; SEPPI, Kevin; LONSDALE, Deryle: Active Learning for Part-of-Speech Tagging: Accelerating Corpus Annotation. In: *LAW'07: Proceedings of the 1st Linguistic Annotation Workshop*, Association for Computational Linguistics, 2007, pp. 101–108.
- [Roy and McCallum 2001] ROY, Nicholas; MCCALLUM, Andrew: Toward Optimal Active Learning through Sampling Estimation of Error Reduction. In: *ICML'01: Proceedings of the 18th International Conference on Machine Learning*, Morgan Kaufmann, 2001, pp. 441–448.
- [Rubens and Sugiyama 2006] RUBENS, Neil; SUGIYAMA, Masashi: Coping with active learning with model selection dilemma: Minimizing expected generalization error. In: *IBIS'06: Proceedings of Workshop on Information-Based Induction Sciences*, 2006, pp. 310–315.
- [Saar-Tsechansky and Provost 2004] SAAR-TSECHANSKY, Maytal; PROVOST, Foster: Active Sampling for Class Probability Estimation and Ranking. In: *Machine Learning* 54 (2004), No. 2, pp. 153–178.
- [Salganicoff et al. 1996] SALGANICOFF, Marcos; UNGAR, Lyle; BAJCSY, Ruzena: Active Learning for Vision-Based Robot Grasping. In: *Machine Learning* 23 (1996), No. 2, pp. 251–278.

- [Scheffer and Wrobel 2001] SCHEFFER, Tobias; WROBEL, Stefan: Active learning of partially hidden markov models. In: *Proceedings of the ECML/PKDD Workshop on Instance Selection*, Springer, 2001 (Lecture Notes in Artificial Intelligence).
- [Schein 2005] SCHEIN, Andrew I.: *Active learning for logistic regression*, University of Pennsylvania, Dissertation, 2005.
- [Schohn and Cohn 2000] SCHOHN, Greg; COHN, David: Less is More: Active Learning with Support Vector Machines. In: *ICML'00: Proceedings of 17th International Conference on Machine Learning*, Morgan Kaufmann, 2000, pp. 839–846.
- [Scholz 2007] SCHOLZ, Martin: *Scalable and accurate knowledge discovery in real world databases*, Department of Computer Science, University of Dortmund, Dissertation, 2007.
- [Schütze et al. 2006] SCHÜTZE, Hinrich; VELIPASAOGLU, Emre; PEDERSEN, Jan: Performance thresholding in practical text classification. In: *CIKM'06: Proceedings of the 15th ACM international conference on Information and knowledge management*, ACM, 2006, pp. 662–671.
- [Schölkopf and Smola 2002] SCHÖLKOPF, Bernhard; SMOLA, Alexander: *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.
- [Settles 2009a] SETTLES, Burr: *Active Learning for NLP: Past, Present, and Future*. 2009. – Invited Talk at the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing.
- [Settles 2009b] SETTLES, Burr: Active Learning Literature Survey / University of Wisconsin–Madison. 2009 (1648). – Computer Sciences Technical Report.
- [Settles and Craven 2008] SETTLES, Burr; CRAVEN, Mark: An analysis of Active Learning strategies for sequence labeling tasks. In: *EMNLP'08: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2008, pp. 1069–1078.
- [Settles et al. 2008] SETTLES, Burr; CRAVEN, Mark; FRIEDLAND, Lewis: Active Learning with real annotation costs. In: *Proceedings of the NIPS 2008 Workshop on Cost-Sensitive Machine Learning*, MIT Press, 2008.
- [Settles et al. 2007] SETTLES, Burr; CRAVEN, Mark; RAY, Soumya: Multiple-Instance Active Learning. In: *NIPS'07: Advances in Neural Information Processing Systems*, MIT Press, 2007, pp. 1289–1296.

- [Seung et al. 1992] SEUNG, Sebastian; OPPER, Manfred; SOMPOLINSKY, Haim: Query by committee. In: *COLT'92: Proceedings of the 5th Annual Workshop on Computational Learning Theory*, Kluwer Academic Publishers, 1992, pp. 287–294.
- [Shen et al. 2004] SHEN, Dan; ZHANG, Jie; SU, Jian; ZHOU, GuoDong; TAN, Chew L.: Multi-Criteria-based Active Learning for Named Entity Recognition. In: *ACL'04: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Morgan Kaufmann, 2004, pp. 589–596.
- [Sheng and Ling 2007] SHENG, Victor; LING, Charles: Partial example acquisition in cost-sensitive learning. In: *KDD'07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2007, pp. 638–646.
- [Shimodaira 2000] SHIMODAIRA, Hidetoshi: Improving predictive inference under covariate shift by weighting the log-likelihood function. In: *Journal of Statistical Planning and Inference* 90 (2000), No. 2, pp. 227–244.
- [Son et al. 2009] SON, Jeong-Woo; SONG, Hyun-Je; PARK, Seong-Bae; PARK, Se-Young: Coping with Distribution Change in the Same Domain Using Similarity-Based Instance Weighting. In: *Advances in Machine Learning* 5828 (2009), pp. 354–366.
- [Symons et al. 2006] SYMONS, Christopher; SAMATOVA, Nagiza; KRISHNAMURTHY, Ramya; PARK, Byung; UMAR, Tarik; BUTTLER, David; CRITCHLOW, Terence; HYSOM, David: Multi-Criterion Active Learning in Conditional Random Fields. In: *ICTAI'06: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, IEEE, 2006, pp. 323–331.
- [Tang et al. 2001] TANG, Min; LUO, Xiaoqiang; ROUKOS, Salim: Active learning for statistical natural language parsing. In: *ACL'02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2001, pp. 120–127.
- [Team 2008] TEAM, R Development C.: *R: A Language and Environment for Statistical Computing* R Foundation for Statistical Computing, 2008. – <http://www.R-project.org>.
- [Tjong Kim Sang and De Meulder 2003] TJONG KIM SANG, Erik; DE MEULDER, Fien: Introduction to the CoNLL-2003 Shared Task: Language-independent named entity recognition. In: *CoNLL-03: Proceedings of the 7th Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2003, pp. 142–147.

- [Tomanek and Hahn 2008] TOMANEK, Katrin; HAHN, Udo: Approximating Learning Curves for Active-Learning-Driven Annotation. In: *LREC'08: Proceedings of the 6th International Language Resources and Evaluation*, European Language Resources Association, 2008, pp. 1319–1324.
- [Tomanek and Hahn 2009a] TOMANEK, Katrin; HAHN, Udo: Reducing Class Imbalance During Active Learning for Named Entity Annotation. In: *K-CAP'09 — Proceedings of the 5th International Conference on Knowledge Capture*, ACM, 2009, pp. 105–112.
- [Tomanek and Hahn 2009b] TOMANEK, Katrin; HAHN, Udo: Semi-Supervised Active Learning for Sequence Labeling. In: *ACL/IJCNLP'09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Association for Computational Linguistics, 2009, pp. 1039–1047.
- [Tomanek and Hahn 2009c] TOMANEK, Katrin; HAHN, Udo: Timed Annotations — Enhancing MUC7 Metadata by the Time It Takes to Annotate Named Entities. In: *Proceedings of the 3rd Linguistic Annotation Workshop*, Association for Computational Linguistics, 2009, pp. 112–115.
- [Tomanek and Hahn 2010] TOMANEK, Katrin; HAHN, Udo: Annotation Time Stamps — Temporal Metadata from the Linguistic Annotation Process. In: *LREC'08: Proceedings of the Seventh International Language Resources and Evaluation*, European Language Resources Association, 2010.
- [Tomanek et al. 2009] TOMANEK, Katrin; LAWS, Florian; HAHN, Udo; SCHÜTZE, Hinrich: On Proper Unit Selection in Active Learning: Co-Selection Effects for Named Entity Recognition. In: *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, Association for Computational Linguistics, 2009, pp. 9–17.
- [Tomanek and Olsson 2009] TOMANEK, Katrin; OLSSON, Fredrik: A Web Survey on the Use of Active Learning to support Annotation of Text Data. In: *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, Association for Computational Linguistics, 2009, pp. 45–48.
- [Tomanek et al. 2007a] TOMANEK, Katrin; WERMTER, Joachim; HAHN, Udo: An approach to text corpus construction which cuts annotation costs and maintains corpus reusability of annotated data. In: *EMNLP-CoNLL'07: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning*, Association for Computational Linguistics, 2007, pp. 486–495.

- [Tomanek et al. 2007b] TOMANEK, Katrin; WERMTER, Joachim; HAHN, Udo: Efficient Annotation with the Jena ANnotation Environment (JANE). In: *Proceedings of the 1st Linguistic Annotation Workshop*, Association for Computational Linguistics, 2007, pp. 9–16.
- [Tong and Chang 2001] TONG, Simon; CHANG, Edward: Support vector machine active learning for image retrieval. In: *Proceedings of the 9th ACM International Conference on Multimedia*, ACM, 2001, pp. 107–118.
- [Tong and Koller 2000] TONG, Simon; KOLLER, Daphne: Support Vector Machine Active Learning with Applications to Text Classification. In: *Journal of Machine Learning Research*, MIT Press, 2000, pp. 999–1006.
- [Triantaphyllou 2000] TRIANTAPHYLLOU, Evangelos: *Multi-Criteria Decision Making Methodologies: A Comparative Study*. Vol. 44. Kluwer Academic, 2000.
- [Tsuboi et al. 2009] TSUBOI, Yuta; KASHIMA, Hisashi; HIDO, Shohei; BICKEL, Steffen; SUGIYAMA, Masashi: Direct Density Ratio Estimation for Large-scale Covariate Shift Adaptation. In: *ICDM’08: Proceedings of the SIAM International Conference on Data Mining*, IEEE Computer Society, 2009, pp. 443–454.
- [Tür et al. 2005] TÜR, Gökhan; HAKKANI-TÜR, Dilek; SCHAPIRE, Robert: Combining active and semi-supervised learning for spoken language understanding. In: *Speech Communication* 45 (2005), No. 2, pp. 171–186.
- [Vijayakrishna and Sobha 2008] VIJAYAKRISHNA, R.; SOBHA, L.: Domain Focused Named Entity Recognizer for Tamil Using Conditional Random Fields. In: *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*, Asian Federation of Natural Language Processing, 2008, pp. 59–66.
- [Vijayanarasimhan and Grauman 2009] VIJAYANARASIMHAN, Sudheendra; GRAUMAN, Kristen: What’s it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In: *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 0 (2009), pp. 2262–2269.
- [Vlachos 2004] VLACHOS, Andreas: *Active learning with Support Vector Machines*, School of Informatics, University of Edinburgh, Master Thesis, 2004.
- [Vlachos 2008] VLACHOS, Andreas: A stopping criterion for active learning. In: *Computer Speech and Language* 22 (2008), No. 3, pp. 295–312.
- [Wang et al. 2009] WANG, Tian-Jiang; CHEN, Gang; HERRERA, Perfecto: Music retrieval based on a multi-samples selection strategy for support vector machine

- active learning. In: *SAC'09: Proceedings of the 2009 ACM symposium on Applied Computing*, ACM, 2009, pp. 1750–1751.
- [Witten and Frank 2005] WITTEN, Ian; FRANK, Eibe: *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd. Morgan Kaufmann, 2005.
- [Wu et al. 2003] WU, Youzheng; ZHAO, Jun; XU, Bo: Chinese Named Entity Recognition Combining Statistical Model with Human Knowledge. In: *Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition*, Association for Computational Linguistics, 2003, pp. 65–72.
- [Yarowsky 1995] YAROWSKY, David: Unsupervised word sense disambiguation rivaling supervised methods. In: *ACL'95: Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 1995, pp. 189–196.
- [Zadrozny 2004] ZADROZNY, Bianca: Learning and evaluating classifiers under sample selection bias. In: *ICML'04: Proceedings of the 21st international conference on Machine learning*, ACM, 2004, pp. 114.
- [Zhang and Kim 1997] ZHANG, Byoung-Tak; KIM, Sung-Hoon: An evolutionary method for active learning of mobile robot path planning. In: *IEEE International Symposium on Computational Intelligence in Robotics and Automation (1997)*, pp. 312.
- [Zhang and Oles 2000] ZHANG, Tong; OLES, Frank: A probability analysis on the value of unlabeled data for classification problems. In: *ICML'00: Proceedings of the 17th International Conference on Machine Learning*, Morgan Kaufmann, 2000, pp. 1191–1198.
- [Zhao and Liu 2008] ZHAO, Jun; LIU, Feifan: Product named entity recognition in Chinese text. In: *Language Resources and Evaluation* 42 (2008), No. 2, pp. 197–217.
- [Zhou et al. 2005] ZHOU, GuoDong; SU, Jian; YANG, Lingpeng: Resolution of Data Sparseness in Named Entity Recognition Using Hierarchical Features and Feature Relaxation Principle. In: *CICLing'05: Proceedings of 6th Conference on Intelligent Text Processing and Computational Linguistics*, Springer Berlin/Heidelberg, 2005, pp. 750–761.
- [Zhu and Hovy 2007] ZHU, Jingbo; HOVY, Eduard: Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem. In: *EMNLP-CoNLL'07: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, 2007, pp. 783–790.

- [Zhu et al. 2008a] ZHU, Jingbo; WANG, Huizhen; HOVY, Eduard: Learning a Stopping Criterion for Active Learning for Word Sense Disambiguation and Text Classification. In: *IJCNLP'08: Proceedings of the International Joint Conference on NLP*, Asian Federation of Natural Language Processing, 2008, pp. 366–372.
- [Zhu et al. 2008b] ZHU, Jingbo; WANG, Huizhen; HOVY, Eduard: Multi-Criteria-Based Strategy to Stop Active Learning for Data Annotation. In: *Coling'08: Proceedings of the 22nd International Conference on Computational Linguistics*, Coling 2008 Organizing Committee, 2008, pp. 1129–1136.
- [Zhu et al. 2008c] ZHU, Jingbo; WANG, Huizhen; YAO, Tianshun; TSOU, Benjamin: Active Learning with Sampling by Uncertainty and Density for Word Sense Disambiguation and Text Classification. In: *Coling'08: Proceedings of the 22nd International Conference on Computational Linguistics*, Coling 2008 Organizing Committee, 2008, pp. 1137–1144.