
Automatic Methods to Extract Latent Meanings in Large Text Corpora

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

der Technischen Universität Dortmund

an der Fakultät für Informatik

von

Christian Pölitz

Dortmund

2016

Tag der mündlichen Prüfung: 24. 10. 2016

Dekan: Prof. Dr.-Ing. Gernot A. Fink

Gutachter: Prof. Dr. Katharina Morik

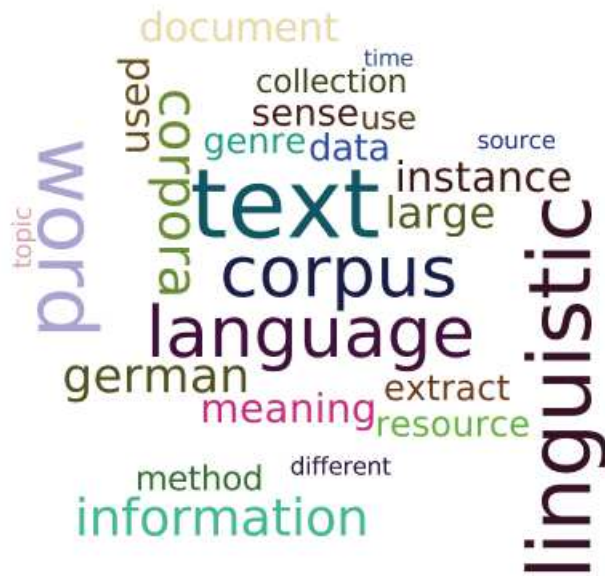
Prof. Dr. Heinrich Müller

Contents

1. Introduction	7
1.1. Impact in Computer Science	9
1.2. Contributions	10
1.3. Corpus Linguistics	11
1.3.1. Evolution of Digital Corpora	12
1.3.2. Retrieval and Concordances	14
1.3.3. Additional Language Resources	16
1.4. Empirical Extraction of Meanings in Corpora	17
1.4.1. Diachronic Linguistics and Variety Linguistics	18
1.5. Pre-studies in Corpus Linguistics	18
1.6. Outline	19
2. Latent Variable Models	23
2.1. Overview	32
2.2. Factor Models	34
2.2.1. Latent Semantic Analysis	34
2.2.2. Partial Least Squares	38
2.2.3. Non-negative Matrix Factorization	39
2.2.4. Kernel Principal Component Analysis	41
2.2.5. Kernel Partial Least Squares	43
2.3. Topic Models	44
2.3.1. Probabilistic Latent Semantic Analysis	45
2.3.2. Latent Dirichlet Allocation	48
2.3.3. Further Topic Models	63
3. Evaluation Methods	65
3.1. Qualitative Evaluation Methods	65
3.1.1. Ranking Lists and Word Clouds	66
3.1.2. Temporal Distribution	68
3.1.3. Geometric Interpretation	70
3.2. Quantitative Evaluation Methods	71
3.2.1. Coherence Measures	71
3.2.2. Likelihood	74
4. Regularized Latent Variable Models	81
4.1. Overview	82
4.2. Topic Models with Regularization	83

4.3.	Factor Models with Regularization	86
5.	Use Case Diachronic Linguistics	93
5.1.	Motivation	93
5.2.	Related Work in Temporal Topic Modeling	95
5.3.	Topic Models for Diachronic Linguistic Tasks	96
5.3.1.	Temporal Topic Modeling	96
5.3.2.	Hierarchical LDA over Time	102
5.4.	Evaluation	106
5.5.	Qualitative Results	107
5.5.1.	Lexicography using the German Reference Corpus	107
5.5.2.	Semantics in Wikipedia Discussions	111
5.5.3.	Semantics in the Spiegel Magazine	111
5.6.	Quantitative Results	114
5.7.	HLDA with Attention Curves	115
5.8.	Conclusion	117
6.	Use Case Non-Standard Corpora Corpus Linguistics	121
6.1.	Motivation	122
6.2.	Dirichlet Priors	122
6.3.	Related Work on LDA with Additional Features	125
6.4.	Regularizing Topic Models by Priors	126
6.5.	Evaluation	131
6.6.	Results	132
6.7.	Conclusion	133
7.	Use Case Variety Linguistics	135
7.1.	Motivation	135
7.2.	Projection-based Regularized Factor Models	137
7.2.1.	Related Work on Domain Adaptation	139
7.2.2.	Moment Matching	140
7.2.3.	Online Distribution Matching	141
7.2.4.	Related Manifold Methods	144
7.3.	Regularized Non-linear Factor Models for Variety Linguistics	145
7.3.1.	Approximating Kernels via Random Features	145
7.3.2.	Non-linear Factors by Distribution Matching	146
7.4.	Evaluation	148
7.5.	Qualitative Results	149
7.5.1.	Lexicographic Varieties across Social Media and Press Media	149
7.5.2.	Semantic Varieties in Social Media	152
7.6.	Quantitative Results	155
7.6.1.	Linear Domain Adaptation	156
7.6.2.	Non-linear Domain Adaptation	163
7.7.	Conclusion	165

8. Software and Integration	169
8.1. RapidMiner	169
8.2. Corpus Linguistics Plugin	170
8.2.1. Interface to Linguistic Resources	171
8.2.2. Text Processing	172
8.2.3. Latent Topic Models	173
8.2.4. Latent Factor Models	175
8.2.5. Evaluation	176
8.2.6. Results	177
8.3. Diachronic Linguistics Process	178
8.4. Variety Linguistics Process	179
8.5. Software in Application	180
8.5.1. Integration into WebLicht	180
8.5.2. Integration into DWDS	181
9. Summary and Conclusion	183
9.1. Summary	183
9.2. Conclusion and Impact	184
9.2.1. Impact	184
9.2.2. Outlook	185
A. Appendix	187
A.1. Collaborations	187
A.1.1. Chapter 1	187
A.1.2. Chapter 5	187
A.1.3. Chapter 7	187
A.2. Publications	187
A.2.1. Accepted Papers	188
A.2.2. Papers under Review	189
A.3. Operator Reference	189
A.3.1. Data Imports	189
A.3.2. Latent Topic Models	191
A.3.3. Latent Factor Models	195
A.3.4. Evaluation Methods	195



1. Introduction

Language has always drawn the attention of people. “I do not know any person, who is not interested in language”. This is a quote from Steven Pinker who wrote one of the most influential commercial books about language: *The language instinct* [Pin94]. In linguistics, researchers investigate and study language in all its aspects. Understanding language is the key to understand the history and the development of mankind. From the beginning of language studies, major directions in the field of linguistics have developed. The two most general fields in linguistics are spoken and written language. Especially written language has been investigated extensively over the last centuries.

Written language enables to store and to transmit information among people, over distance and time. This decouples the human evolution from purely environmental influences to knowledge-based influences. Before we started to keep track of our experiences as stories, poems or religious texts, only a very limited amount of knowledge could be kept within the generations of men. Preserving knowledge by writing makes human achievements and experiences available

1. Introduction

for future generations.

The language used in written records can be used to extract more than pure information. The context, writing style or used language elements is investigated to understand meaning and intention of written records. For this purpose, whole collections of documents and texts are compiled to so called corpora (singular: corpus). A (text) corpus is a collection of texts that is compiled for certain analytical tasks. It is generally not simple to decide if a collection of texts is a corpus or just a collection of texts. The easiest way to distinguish a corpus from a simple text collection is by the intention behind the compilation. If documents or texts are collected to perform analyses, we could speak about a corpus for example. If texts are only collected for storage, we could speak about document collections. In [Hun06], Susan Hunston describes a corpus as collection of documents helping linguistic researchers to perform certain tasks. A corpus is coherent to accomplish, to validate and to extract linguistic hypotheses. On the other hand, a corpus contains variance to accomplish a broad variety of possible hypotheses to validate or extract. Throughout this thesis we use the notation of a document or text collection to describe compiled written records that are used for any analysis (linguistic or not). If we refer to certain linguistic tasks, we use the notation of a corpus.

One of the oldest corpora used for linguistics is the Bible. Already 1790, Alexander Cruden published "A Complete Concordance to the Holy Scriptures" [Cru06]. In this book, the author compiled a collection of all words from the King James Bible including references and information about the use. For centuries, the availability of the Bible for such linguistic analyses was only possible for a few cleric persons. Before the first German translation of the Bible by Martin Luther (1522) for example, most people could not read the Bible at all.

Prior to the dawn of modern computers to become available for linguistic and language experts, analyses on corpora were done by hand. With increasing numbers of documents available, systematic analysis of the language in the texts became possible. The field of corpus linguistics uses corpora for systematic linguistic analysis. While classical corpus linguistic studies are concentrated on small text collections to perform qualitative linguistic tasks, quantitative methods based on modern statistical and automatic analysis methods enable to perform large analyses on bigger text corpora.

From the 1960s on, modern digital text corpora offer large text collections like newspaper articles, social media content, but also language reference corpora. Since the beginning of the Internet, more and more textual information has become available for everybody. With the availability of such large collections of digital documents, electronic document collections and corpora start to be used for linguistic analysis. To use such large amounts of digital texts, non-manual methods to extract information for linguistic research become more important. Natural Language Processing (NLP) methods [MS99] for example can help to automatically analyze large document collections and corpora. In NLP we try to discover knowledge from language data sources. NLP uses methods that perform automatic analysis tasks based on identification of patterns in texts. The goal is to find information in the data when manual analyses are not possible, too expensive or too time-consuming.

Two major sub-fields of linguistics can extraordinarily benefit from such automatic data driven analyses as being offered by NLP methods. First, in lexicography the uses of words and expres-

sions is studied. In dictionaries like the Cambridge dictionary¹, different usages and senses of words are collected. To keep these information up to date and enrich them with examples for the individual usages and sense, large text corpora are used. In [BRP14, LPDG] and [GPB], we show how useful NLP methods are to assign a large number of documents to possible senses from dictionary entries.

Second, the meaning of texts and parts of texts is investigated to infer the writers intention. In sentiment analysis or analysis of Internet-based communication for instance, we want to extract sentiments or intentions of the writers from reviews or chat entries. The field of semantics concentrates on studying meanings. Common large digital text corpora do not distinguish between different meanings of word forms, intense manual effort has to be done for disambiguating texts. Automatically disambiguating texts from large digital corpora by NLP methods is proven to be effective in our works in [PB14b, BLMP14, PB14a] and [BPMS14].

In both fields, our investigations show that we are actually extracting latent aspects from the documents in the corpora that can be associated with meanings. Latent means that we do not directly see the meaning, we need to infer it from context. Meanings in this context means a significant usage of words in a document for a certain purpose. For example the word "bank" can be associated with the possible meanings from a dictionary:

- a financial institute, or
- a land along some water

From an example sentence like "I was standing at the bank of the river Thames all day long.", we do not directly know which meaning fits best. But, considering accompanying words, we can infer the latent concept behind the use of the word and map this to a meaning. All this results in an extraction of the hidden concepts from the documents to infer meanings.

1.1. Impact in Computer Science

So far, we motivated this thesis for the benefits in corpus linguistics. Next, we describe the significance of this thesis in computer science. Corpus linguistics on digital text corpora is a special research fields of Natural Language Processing (NLP). It is one of the oldest fields in computer science. Starting with Alan Turing's test about computer intelligence in 1950 [Tur50], NLP was thriving force of new computer science methods as well as application. The earliest approaches in NLP are from the fields of machine translation of different languages and automatic syntactic annotations. From the late 1960s, computer linguistic conferences like the Association for Computational Linguistics (ACL) and the Conference on Computational Linguistics (Coling) present linguistic research in these fields using computer science (cf. [Cul65, ST69]). Already in 1965, computer linguistics investigated [PD65] automatic learning for linguistic classifications. Since then, data analysis and artificial intelligence methods provide a large range of sophisticated methods in NLP. Named Entity Recognition [CMP03], Sentiment Analysis [PL08]

¹<http://dictionary.cambridge.org/>

1. Introduction

or Word Sense Disambiguation [Nav09] are only a few examples of how modern NLP or general automatic analyses methods are applied in linguistics. In these areas, new computer science methods emerged due to the need of automatic language processing.

The methods that we develop in this thesis and the studies we perform, contribute to these NLP methods for computer science. Especially in the last years with the enormous growth of digital texts available from social medias, search engines or online applications, methods to analyze (written) language become more and more important. Facebook for example uses NLP for automatic text understanding from user communications². Researchers and engineers face similar challenges in NLP as we do in corpus linguistics for social media analysis. Such analyses on social media content can be supported by the methods and studies performed in this thesis. Especially, large scale statistical models based on artificial intelligence methods help companies like Facebook to understand their users. Our developments on data analysis for diachronic linguistics and variety linguistics are easily applicable for text understanding in social media. Varieties on language of the users and changes of user behavior over time are present in Facebook's user content in the same way as they are in the corpora that we investigate in this thesis. Further big IT companies like Google or Apple need techniques and analysis methods for text as the ones we develop in this thesis. For Google, NLP was important from the beginning to support their search engine. Currently, both Google and Apple highly bet on Language Processing for Human Computer Interactions (HCI)³. Both companies possess large amounts of written text from their users. Whether query logs from Google or user data from Apple, these big text collections offer large potentials for linguistic analysis that support HCI systems.

1.2. Contributions

The main contribution of this thesis is the improvement of the performance of empirical linguistic tasks on large digital document collections and digital corpora with automatic analysis methods. Available heterogeneous and structured language resources provide a plethora of textual, statistical or expert information for linguistic research. In this environment, the use of modern NLP methods to support linguistic studies is investigated. Based on large digital corpora, novel approaches to efficiently perform linguistic tasks in lexicography and semantics are developed.

Contribution in Corpus Linguistics

Current research in corpus linguistics uses modern digital corpora only to search and retrieve samples from documents. These samples are usually manually processed further. Such post-processing includes annotating texts and words with syntactic and semantic information in context. So far, automatic computer-aided methods have been applied to infer syntactic information

²<https://code.facebook.com/posts/181565595577955/introducing-deeptext-facebook-s-text-understanding-engine/>

³<http://www.forbes.com/sites/jaysondemers/2015/08/12/how-far-can-googles-linguistic-analysis-go/#1290f0516eba>

like word classes (verb, noun, etc.). The methods developed in this thesis provide additional annotations of semantic categories for diachronic and variety linguistic tasks by automatic analysis methods in the fields of lexicography and semantics.

The identification of semantic categories in corpora is done by latent variable methods. We accomplish interpretability of results by providing associations of the latent variables to words. This is in strong contrast to black box approaches as offered by Neural Networks and current Deep Learning approaches. In pre-studies, a list of words and document collections that are of interest in (computer) linguistic research is compiled. These words are expected to show differences in use depending on time and text genre. On large scale use cases, the benefit of the proposed methods are shown.

Finally, a tool to perform linguistic tasks on large scale digital corpora is developed. This makes the methods available for linguistic research and teaching.

Contribution in Computer Science

The contribution to computer science research are inherent in the methods developed. First, Chapter 4 of this thesis unifies latent variable methods that use additional information about the data and regularized optimizations. This results in a common framework for latent variable methods with additional information. For evaluating the latent variable methods, Chapter 3 discusses methods that estimate the quality of temporal topic models with Sequential Monte Carlo methods. A novel Monte Carlo method for the estimation of joint and conditional likelihoods of temporal topic models is developed. A new temporal coherence measure that compares the modeled time from a temporal topic model with the empirical distribution of the time stamps in a corpus is proposed.

The methods in the use cases are non-trivial extensions of latent variable methods that allow efficient and effective extraction of hidden information from large data collections with additional heterogeneous data resources. In Chapter 5, latent variable models for diachronic linguistics are introduced. These models extend standard latent topic models with a temporal distribution that is based on diffusion processes. Compared to previous approaches for temporal topic models, we model an attention process that more precisely covers temporal distributions in document collections. Chapter 6 introduces effective methods to integrate word information from heterogeneous language resources to extract latent information from corpora with scarce and sparse data. Via lasso and group lasso regularization, background information about words are integrated into the extraction of latent meanings in digital corpora. In Chapter 7, an optimization problem is set up for variety linguistics that is a novel method to efficiently extract latent subspaces that model similarities and dissimilarities of large data collections. An efficient method is elaborated to perform an extraction of latent factors by an optimization on a matrix manifold.

1.3. Corpus Linguistics

Before the technical foundations of this thesis, we give motivation from the linguistic point of view. We will describe corpus linguistics as field that extraordinarily benefits from NLP meth-

1. Introduction

ods. In a pre-study we show how automatic analysis methods help supporting corpus linguistic tasks.

A (text) corpus is a collection of text documents that has been assembled for a special purpose. In linguistics, such corpora are used to perform linguistic analyses or validate linguistic hypotheses. The field of corpus linguistics studies language based on such corpora. There exist not only text corpora, but also corpora of spoken language for example. In this thesis, we concentrate on text corpora. Modern digital corpora offer a large number of pre-processed text documents of different genres and sources for linguistic tasks. In recent years, language resources have been built, including large digital corpora. The Common Language Resources and Technology Infrastructure (Clarin)⁴ for example establishes a collection of language resources around Europe.

Interfaces accomplish efficient access to the texts. This access can be on document level or on snippet level. Document level access allows for retrieval of whole documents including meta information like source, genre, time of publication or authorship. Snippet level access allows only for retrieval of document parts. These parts are usually identified by linguistic queries. A linguistic query is analogue to queries for modern search engines, but offer additional linguistic features.

Besides the documents in a digital corpus, modern linguistic infrastructures offer external information and statistics about the documents and words in the corpus. Language resources like the Dictionary of the German Language [Gey07] provide large text corpora and interfaces to extract text snippets as result of linguistic queries, see Figure 1.3. Such queries ask, for instance, for all occurrences of a certain word. Additional constraints like considerations of lemmas or Parts-of-Speech can be used as complex linguistic queries. The results of such queries are so called KWIC-lists. These Key-Word-in-Context (KWIC)-lists contain text snippets of small contexts that match the corresponding query. Given a KWIC-list for a given key word query, the snippets are used for different linguistic tasks. Additional to the KWIC-lists, we can extract several information about the corresponding documents and the contained words.

1.3.1. Evolution of Digital Corpora

Since the merit of computers, digital text collections have been compiled for storage and analysis of written language. Similar to [BS06], we distinguish four major periods in the dissemination of digital text corpora:

1. Hand-crafted digital corpora (1960)
2. Scanned texts collections (1990)
3. Internet Content (2000)
4. Social Media Content (now)

From the 1960s on, the Brown Corpus [FK79] has been compiled as one of the first digital corpora. The first version of this corpus contained approximately 1 million running word tokens

⁴<http://clarin.eu/>

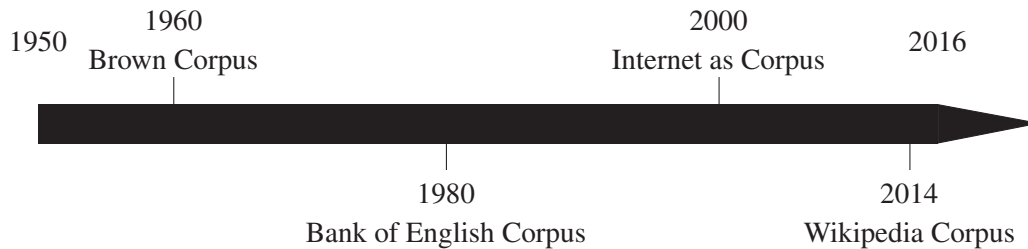


Figure 1.1.: Examples of the development of digital text corpora from the early 60s with hand-crafted text collections (Brown Corpus) to social media content today (Wikipedia).

from publications from different text categories. At this early time, the compilation was done by hand. With the availability of scanners, printed documents could be automatically collected to compose corpora. The Bank of English corpus for example was one of the first digital corpora that was compiled by scanning documents with OCR⁵, see [Jär94] and [BS06]. Since the Internet age, corpora based on World Wide Web content have been used more and more [Hof00]. Further, with the availability of the Internet for almost everyone, the access to large digital corpora by web services becomes the norm [Dav10]. Recently, social media content is used for linguistic analysis. The analysis of Internet-based communication via large collections of chat or blog entries for example becomes an emerging trend in linguistics [BLMP14]. A time line of example corpora over the last 60 years is illustrated in Figure 1.1.

German Corpora

For German linguistics, over the last years several text corpora have been assembled for linguistic research. One of the largest German text corpora is the “Core-Corpus” of the German language from the Dictionary of the German Language, see [Gey07]. This corpus is maintained at the Berlin Brandenburg Academia of Science. The corpus contains 100 million running words, balanced chronologically (over the decades of the 20th century) and by text genre (fiction, newspapers, scientific and functional texts). Additional, newspaper text collections are available to picture the course of the current affairs over time. The newspaper corpus Die ZEIT covers all the issues of the German weekly newspaper Die ZEIT from 1946 to 2009, approximately 460 million running words. An overview of the available text collections can be found on the home page of the Dictionary of the German Language: www.dwds.de. Another large German text corpus is DeReKo (Deutsches Referenz Korpus) [KK09] provided and maintained by the Institute of the German Language (IDS). Similar to the corpora from the Dictionary of the German Language, DeReKo contains documents from fictional, scientific and newspaper texts. The corpus contains approximately 30 billion running word tokens with additional syntax information.

⁵Optical character recognition: Automatic conversion of images of text into electronic texts.

1. Introduction

Corpus	running words
Brown (first version)	1 million
DeReKo (1992)	28 million
DWDS Core	125 million
Die Zeit	225 million
Bank of English	650 million
Wikipedia (engl. articles)	3 billion
DeReKo (2015)	28 billion

Table 1.1.: Examples of digital corpora and the sizes. The first column names example corpora and the second column tells the number of word with repetition in the corpora.

Development of Digital Corpora

The development of the different digital corpora has led to an increase in available texts for linguistic analyses. While the early corpora like the Brown Corpus consisted of up to 1 million word tokens, modern corpora like the Corpus of Contemporary American English [Dav10] contain already several hundred million word tokens. In Table 1.1, we report the sizes of a number of available digital corpora. In the last years, the size of available digital corpora blew up tremendously. The DeReKo for example started from 28 million [KK09] running words to 28 billion running words by now⁶. In Figure 1.2, we illustrate to growth of DeReKo during the last 15 years.

Besides these corpora, further corpora and document collections from different sources from Internet communications provide a valuable language resource. Social media corpora like the Wikipedia articles and discussion pages [ML14] provide large amounts of texts for linguistic analysis on Internet communication. Further, Amazon reviews [BDP07] about different products provide a large source to compile a social media corpus that enables investigations of Internet communication. Also the 20 newsgroups dataset [Lan95] or the Reuters dataset [LYRL04] are commonly used free document collections as social media corpora.

1.3.2. Retrieval and Concordances

The retrieval of texts from the corpora offered by the different language resources is done via so called KWIC-lists. A KWIC-list contains texts that match a linguistic query - the snippets. Such queries can be single words, phrases or logical expressions. Given for example the query “bank”, we retrieve a list of texts containing this word, see Figure 1.3. Beside this retrieval of KWIC-lists, the whole corpus can also be accessed. In this case we use whole books or documents from the corpus for linguistic tasks. The publicly available language resources usually do not allow to retrieve the whole corpus due to copyright.

Considering KWIC-lists of contexts of a certain word, we also speak about concordances. Concordances list words from corpora with references and possible information about the concrete use. In lexicography, we use concordances since we are interested in the usage of words

⁶<http://www1.ids-mannheim.de/kl/projekte/korpora/archiv.html>

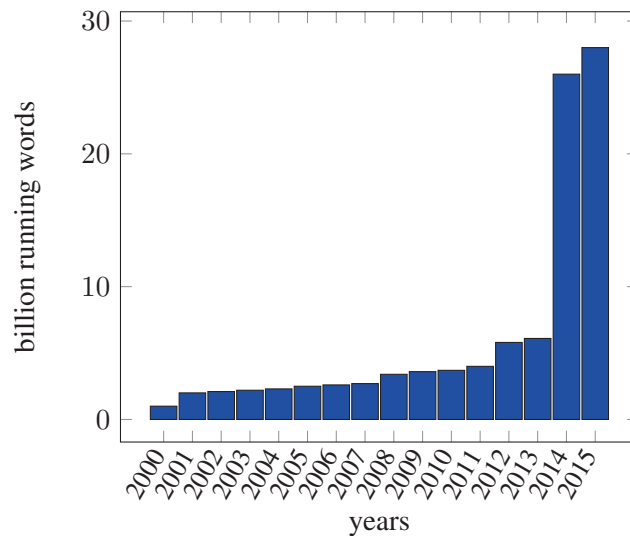


Figure 1.2.: Increase in corpus size for the DeReKo from the Institute of German Language. For the years 2000 to 2016 (on the x-axis), the number of words with repetition (running words) in the DeReKo corpus at these years are plotted on the y-axis.

D W D S

bank

Kernkorpus 20

Treffer: 8409, davon anzeigbar: 7038

KWIC	Datum ↓	Datum ↑	Zufällig	Links	Rechts
1	1999	B	degenhar	Sie saßen manchmal in der Küche beieinander oder auf der Bank unter der Terrasse , lachten , erzählten sich Geschichten .	
2	1999	B	degenhar	Sie saßen nach dem Abendessen auf der Bank vor Andys zur Nordseite , etwas unterhalb des Haupthauses , z	
3	1999	B	degenhar	chen Liegenschaften sowie Beteiligungen an Versicherungen , Banken , Brauereien und Baustoffunternehmen bestehenden Vermöge	
4	1999	B	degenhar	gesehen , wie sie weinte und betete , nach der Beichte , in der Bank kniend , ohne Zweifel erschüttert .	
5	1999	B	degenhar	l daß die zur-Lindens dann durch Einflußnahme auf die beiden Banken einfach eine Kreditsperre gegen die Bramkamps durchgesetzl	
6	1999	B	degenhar	ls nicht von noch so vielen Zur-Linden-Männern , die durch die Bank keine Athleten sind , und nicht mit noch so vielen Zur-Linden-Fr	
7	1999	B	degenhar	rsonal in Umarmung mit einem seiner Witzelleferanten auf der Bank vor der großen Küche gesichtet hatte , mußte man anderen Abw	
8	1999	B	degenhar	sogar draußen hören konnte , wo Nelly und Makewka auf der Bank saßen , sich herumdrehen , die Köpfe schüttelten und sich dan	
9	1999	B	dueckers	Dann setzt sie sich auf die Bank vor Gerts Grab und schließt die Augen .	
10	1999	B	dueckers	ft aus ihrem Spiralheft vor , wenn sie Kuchen essend auf Ihrer Bank sitzt .	
11	1999	B	dueckers	Dann endlich setzt sie sich auf ihre Bank , schenkt Gerts Namen das übliche liebevolle Lächeln und pack	
12	1999	B	dueckers	Ein Mädchen mit kurzen roten Haaren setzt sich auf ihre Bank , blättert in ihrem Buch. Deins ?	
13	1999	B	dueckers	Das Mädchen streckt sich auf der Bank aus , es trägt einen grügelben Trainingsanzug , klobige Turnschl	
14	1999	B	dueckers	Als Rosemarie einen Schritt näher an die Bank herantritt , sieht sie , daß das Mädchen unter seiner grügelben	
15	1999	B	dueckers	Er hat sich zu ihnen mit auf die Bank gesetzt , Yesterday hockt auf seinem Schoß .	

Version: 1.1

Figure 1.3.: For the word “Bank”, example sentences are retrieved from the DWDS Core-Corpus. The examples are called KWIC-lists and contain sentences containing the word “Bank” and the 3 sentences before and after. The figure shows the web interface to retrieve the examples.

1. Introduction

and the extraction of possible meanings or word classes. In semantics we are usually considering whole documents or text snippets that might contain a certain word, but the focus lies on the whole context.

In this thesis, we speak about corpora and documents. For the developed methods, we use only samples from the corpora (as KWIC-lists for example). For notational convenience, we call the collection of these samples also a corpus or simply a text collection. The term document is handled similarly. Although we may use only samples from documents that contain word examples (sentences with context), we call these samples also documents if this notation is clear in the context.

1.3.3. Additional Language Resources

Besides the pure text documents from the corpora, modern language resources as offered by the Dictionary of the German Language, provide additional information about texts and words. For instance, for words we can extract word profiles. Word profiles are statistics of the occurrences of words in a large text corpus and co-occurrences with other words. The statistics can be estimated based on the given corpus or can be provided from an external data source. WordNet [Mil95] for example contains information of hypernyms (the general topic) and hyponyms (more specific subtopics) of words and senses of words.

Meta data and information about the documents in the corpora provide valuable additional features for analyses. Such information can be for instance the author or the publication date of the documents. The snippets from the KWIC-lists have references to the corresponding documents with information about the author, genre and publication date. Additional links between documents within a corpus can be implicitly given by common meta information like the same genre, the same publisher or author. On the other hand explicit links between documents are given in large web corpora, for instance.

Another important language resource are dictionaries. The dictionary of the German language [KG10] or etymological dictionaries provide valuable information about words, like definitions, lemmas, hyponyms or homonyms. This information gives background knowledge about classes and groupings of words. Word nets like GermaNet [HF97] offer information about relations of words based on concepts and semantic relations between them. These word relations are based on ontological and semantic information. Additional co-occurrence information as in the German word profiles [DG13] provide insight about affinity of words in certain relations. Such relations can be words always appearing together if one of them is a direct object for instance. The Dictionary of the German Language provides access to additional language resources accompanying the text corpora via a web interface. For individual words, we can retrieve word profiles or corresponding dictionary entries. See Figure 1.4 for the web interface of the additional language resources offered by the Dictionary of the German Language.

The reason to use additional language sources besides the text corpora is manifold. For example, we might have only small result lists or a small number of documents retrieved from a digital corpus. This can be, for instance, the case for concordances of rare words. Then, we want to use the external data to support the linguistic tasks by information about the words.

1.4. Empirical Extraction of Meanings in Corpora

Figure 1.4.: Web interface to the DWDS language resources. For the word “Ampel”, dictionary entries, synonyms (top) and text examples (bottom) are retrieved via the web interface: www.dwds.de.

1.4. Empirical Extraction of Meanings in Corpora

Extracting meanings from large digital corpora for lexicography and semantics helps maintain- ing dictionaries and investigating language and writing styles. The major concern of this thesis is the extraction of possible meanings of words and documents (snippets) in large digital corpora. In the field of corpus linguistics, automatic extractions of possible meanings help researchers in many linguistic tasks.

We investigate two main use cases that stem from current corpus linguistic research. First, diachronic corpus linguistics investigates the distribution of linguistic phenomena over time. Especially the change of meaning over time is of interest. The German word *Ampel* for example was used in the meaning of a hanging light (on streets for instance) until the second half of the last century. Second, variety linguistics investigates the distribution of linguistic phenomena over different kinds of texts. In sentiment analysis for instance, positive or negative opinions might depend on the context. In product reviews, technical products for instance might be described as well functioning, while books might be described as exciting. Both expressions “well functioning” and “exciting” have positive meanings but are surely differently distributed in technical reviews and book reviews. Both, diachronic linguistics and variety linguistics, find applications in lexicography and semantics.

1. Introduction

1.4.1. Diachronic Linguistics and Variety Linguistics

There are two main applications that can benefit especially from NLP methods on large digital corpora: diachronic and variety linguistics. In diachronic linguistics, we study linguistic phenomena or certain aspects of language over time. The term diachronic is literally translated as “across time”. Diachronic linguistics is also referred to as historical linguistics since it studies language and language changes over time with respect to humanity. In [Byn77], Bynon describes diachronic (or historical) linguistics as: “[...] seeks to investigate and describe the way in which language change or maintain their structure during the course of time.”. Given the German word *Ampel* for example, we have several possible meanings. In its ancient meaning, the word *Ampel* is used as a light. Recently, the term *Ampel* is used in the sense of a coalition of the Social Democratic Party, the Liberal Party and the Green Party. The use of these different senses in a text collection will depend on the writing time of the corresponding document. In this thesis we concentrate only on the language changes in digital text corpora over time. By contrast, the whole field of diachronic linguistics covers more areas like etymology, language families, syntax, morphology or phonology.

In variety linguistics, we study different use and user related varieties in texts. In [Hud96], Hudson defines a variety in linguistics as: “a set of linguistic items with similar distribution”. Such varieties can be for example different sources or text genres that also differ in the use of language. Certain writing styles are clearly source or genre dependent. In fictional literature we can, for instance, expect more figurative speeches and first-person narration. In scientific literature on the other hand, we will find more neutral and passive narrations. The German word *Leiter* for example can be used in the meaning of a ladder or in the meaning of head, director or manager. In fictional literature, we can expect that none of these meanings will be more present in the texts. In newspaper articles on the other hand, we will very likely have more usages of the word *Leiter* in the meaning of head, director or manager. While variety linguistics covers many different areas like dialects, jargons or sociolects, we concentrate on investigations of different genres and document sources in this thesis.

1.5. Pre-studies in Corpus Linguistics

In two studies, we investigate the use of NLP methods for corpus linguistics as part in the *Bundesministerium für Bildung und Forschung (BMBF) Project KobRA*⁷. First, we study how certain words are used across different text genres. Second, we investigate the development of meanings over time. These investigations motivate this thesis. The results will show how useful NLP methods can be for the different linguistic tasks. While these first experiments use state-of-the-art methods, in the thesis we will show how to systematically extend the standard approaches to explicitly adopt to corpora from heterogeneous language resources with information about time and document source (genre).

In both studies, we extract meanings of words that appear together in documents for inferring semantic relations. Individual words and documents can be assigned to the meanings by estimated likelihoods. We analyze how these meanings are distributed over text genres and time.

⁷Korpus-basierte linguistische Recherche und Analyse mit Hilfe von Data-Mining

These studies are reported in the joint publications [PB14a] and [BPMS14]. From the DWDS Core-Corpus, we retrieved a KWIC-list of snippets containing the German word *Platte*. The snippets are drawn from documents from different genres over a time period from 1900 to 1999. The used genres are "Belletristik" (fiction), "Zeitung" (newspaper), "Wissenschaft" (scientific) and "Gebrauchsliteratur" (functional texts).

For these snippets, we extract meanings of words by Latent Dirichlet Allocation [BNJ03] (In the survey chapter, we explain this method in detail). In Figure 1.5, we illustrate two extracted meanings for the word "Platte". At the top, we show the words that are most important for each meaning by a Word Cloud⁸. We see that we extract two different usages of the word *Platte*. First, *Platte* in the meaning of a hard drive is found. The most important words in this group are highly computer related. The second meaning presents the word *Platte* as photographic plate. The important words are all connected to photography. In the middle of the figure, we show the presence of the meanings in different genres by counting how many times the word "Platte" in a document from a certain genre in the corpus has been identified as belonging to one of these meanings. The meaning of hard drive is mostly found in newspaper articles, while the meaning of photography is more present in scientific articles. Here, we see a clear variety of the meanings over the genres. At the bottom, we plot how much present the meanings are over the time in the whole corpus. We count how many times the word "Platte" appears in each meaning in each year in the whole corpus. We see that *Platte* in the context of a hard drive is mostly used at the end of the 20th century, while *Platte* in the context of photography has its climax in use in 1950s.

1.6. Outline

In the next chapters, we present the course of the research from this thesis in corpus linguistics with NLP methods. First, different approaches and state-of-the-art methods for latent variable models are discussed. The foundations of text representation for the application of NLP methods are introduced and we motivate the use of latent variable models to extract hidden concepts from a document collection. In Chapter 2, a detailed introduction to factor and topic models, including mathematical and geometrical principles is given. The methods and methodologies to evaluate latent variable models in qualitative and quantitative ways are described in Chapter 3. After this introduction, we present in Chapter 4 a general outline of the methodology we use to solve corpus linguistic tasks on heterogeneous language resources with large text corpora. In three extensive use cases, we present the results of the methods for corpus linguistics. These use cases are research oriented and proved to be useful for linguistic researches. In the BmBF (Bundesministerium für Bildung und Forschung) project *KobRA*⁹, the significance of these use cases for linguistic research have been shown. For diachronic linguistics, a study on the development of word meanings and subjects in different text collections is presented in Chapter 5. In Chapter 7, we report the results of a study on variety linguistics to compare large text collections by latent factor models. A final study is performed in Chapter 6 to investigate the use of the developed methods for non-standard text collections as they appear in Internet-Based Communication

⁸A Word Cloud visualizes frequent words and their importance by font size

⁹<http://www.kobra.tu-dortmund.de>

1. Introduction

for example. In Chapter 8, we will give a description of the software developed to implement the discussed methods. We explain how the software is used to perform corpus linguistic tasks. Finally, a conclusion of the success of the methods developed in this thesis is drawn and the impact in research and teaching is shown. In the appendix, collaborations and publications in the context of this thesis are discussed. At the end, a summary of the used notations, short references of the used methods and principles together with their acronyms are given in the Glossary.

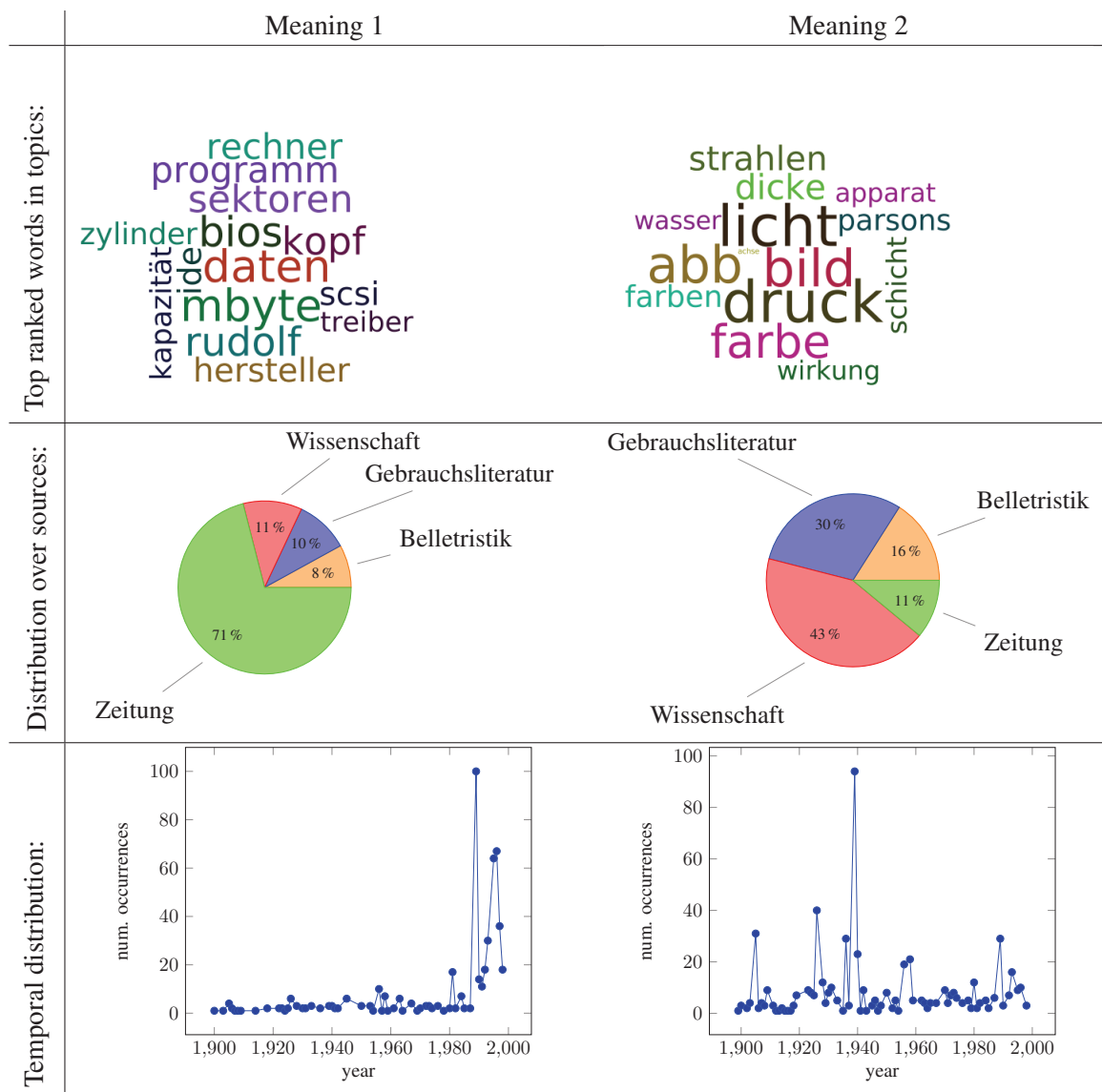


Figure 1.5.: Two meanings extracted from a KWIC-list of snippets for the query *Platte* (plate) from the DWDS Core-Corpus. The first column shows the results for the word “Platte” in the meaning of a hard drive, the second column shows the results for the meaning of a photographic plate. Top: The most important words for the meanings are plotted as word cloud. The larger the word is in size, the more important the word is for this meaning. Middle: The distribution of the meaning over the document sources Wissenschaft (scientific literature), Gebrauchsliteratur (non-fictional literature), Belletristik (fictional literature) and Zeitung (news papers) are shown as pie charts. The pie charts show how many times the meaning of a hard drive or a photographic plate for the word “Platte” can be found in the different sources. Bottom: The temporal distribution of the meanings is plotted. For each year from 1900 to 2000 (x-axis), we count how many times the word “Platte” appears in the two meanings in a document (y-axis).



2. Latent Variable Models

In this chapter, we give the mathematical and methodical background for the thesis. We describe latent factor models that are used in the use case in Chapter 7. Further, latent topic models are explained to pave the ground for the use case in Chapter 5. Finally, we mention previous work on prior distributions for topic models which is the motivation for the use case in Chapter 6.

Analyzing corpora, compiled from large collections of documents, by Natural Language Processing methods helps to automatically and autonomously extract descriptions and summarization of the contained texts. In order to apply such methods, we need to represent the documents in an appropriate way. We distinguish two major approaches for document representation: the Vector Space Model and the Multinomial Model. The Vector Space Model (VSM) represents the documents as vectors in the Euclidean space. In the VSM, we define a basis $\{\mathbf{w}^1, \dots, \mathbf{w}^V\}$ for basis vectors $\mathbf{w}^i \in \mathbb{R}^V$ associated with the words in the vocabulary, with V the number of words in the vocabulary of the corpus. The $\{\mathbf{w}^i\}_{i=1}^V$ are the standard basis vectors in the Euclidean space of dimension V . This means, \mathbf{w}^i is a sparse vector that has only one non-zero

2. Latent Variable Models

entry at the component i with a value of one. Now, each document can be represented as linear combination of these basis vectors. This representation is called the Word-Vector representation of a document in the VSM and is based on the so called Bag-of-Words. The Bag-of-Words (BoW) Model simplifies a document to the set of its words, keeping multiplicity of words but no further structure. We note the Word-Vector of a document d as vector \mathbf{w}_d . Given for example a corpus of only the document “ $a b a c$ ”. The BoW representation is: $d = \{a, b, c\}$. The corresponding VSM uses three basis vectors of dimension three:

$$\mathbf{w}^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{w}^2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{w}^3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

The first basis vector is associated with the first word from the vocabulary (a), the second basis vectors is associated with the second word (b) and the third basis vector with the third word (c). Using these basis vectors, we can now write the document as a linear combination

$$\mathbf{w}_d = a_{1d}\mathbf{w}^1 + a_{2d}\mathbf{w}^2 + a_{3d}\mathbf{w}^3.$$

In the literature several ways to define this linear combination have been proposed. The simplest approach is to define the weights a_{id} as the multiplicity of word i in the corresponding document d . This results in the following Word-Vector that collects the number of occurrences of the words in the document:

$$\mathbf{w}_d = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

This representation considers only the number of times a word appears in a document, but does not consider how many words the document has. The relative frequency of the words can be used to explicitly model the importance of a word w_i in document d by a weight value: $\frac{n_{d,w_i}}{\sum_{w_j \in d} n_{d,w_j}}$, with n_{d,w_i} the number of times word w_i appears in document d . This results in the following Term Frequency (TF) Word-Vectors of the document from above:

$$\begin{bmatrix} 0.5 \\ 0.25 \\ 0.25 \end{bmatrix} = 0.5 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 0.25 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 0.25 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Further weightings for the Word-Vectors are possible. The most prominently used weights are the so called Term Frequency Inverse Document Frequency. The TF-IDF-values additionally weight the term frequency by the inverse document frequency. The inverse document frequency is the logarithm of the inverse frequency of the word in all documents. By this, words that appear in many documents are weighted down. For different learning tasks such weighted Word-Vectors have proven to be more representative for the words in a document. See [SB88] for a discussion on different weights for terms.

An advantage of the VSM representation is that it allows for a geometric interpretation of the documents and the collection of documents from the corpus: In the VSM, a document is represented by a geometric object (the Word-Vector) in the Euclidean space. The document content stems from a combination of the geometric objects (the basis vectors) in the space. Further, we can calculate distances between documents to estimate similarities between documents and words.

Given a corpus as a document collection $\{d_1, \dots, d_M\}$, we can collect all Word-Vectors $\{\mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_M}\}$ into a whole matrix X . This Term-Document Matrix contains the Word-Vectors as columns:

$$X = [\mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_M}].$$

The matrix can be used to describe the document collection as a single object. We can use Linear Algebra to model the space spanned by the Word-Vectors and the space spanned by the rows of the term document matrix. The first space can be seen as description of the documents in the collection based on the contained words. This space is spanned by the basis vectors $\mathbf{w}^i \in \mathbb{R}^V$ associated with each word. Every column vector of X lies in this space. The second space can be seen as description of the words in the collection based on the documents they appear in. This space is spanned by basis vectors associated with the documents. Every row vector of X lies in this space. Later we will explain how this representation of the collection can be used to extract properties of the corpus.

Another way to represent documents is the Multinomial Model (MM). The MM models documents as sequence of words drawn from a multinomial distribution. The multinomial distribution is a discrete probability distribution over a number of objects. In our case, the objects are the words w_i from the vocabulary. Each word has probability $p(w_i)$ of occurring in a document. The document is assumed to be generated by a random process that generates a sequence of words which each word generated with probability $p(w_n)$. Hence, the documents in the corpus are represented as sequences of realizations of random variables - the words: $\mathbf{d} = (w_1, \dots, w_N)$. The probability distribution of such a sequence is modeled by a statistical Language Model (LM) - see [PC98]. A LM is a model that estimates the probability of a sequence of words.

The individual probabilities need to be estimated from the corpus. Depending on the LM, we estimate the marginal distributions $p(w_n)$ and additional conditional distributions $p(w_n | w_1, \dots, w_{n-1})$ based on counts of the appearances of the words in the documents.

If we assume an uni-gram LM, each word is independent of all other words in a document. The probabilities $p(w_n)$ can be derived by Maximum Likelihood Estimation (MLE). In MLE, the probabilities are estimated that maximize the likelihood of the data given these probabilities. An introduction into MLE can be found in [Myu03].

Using MLE, we estimate the word probabilities

$$p(w_n) = \frac{n_{w_n}}{\sum_j n_{w_j}},$$

for n_{w_n} the number of time word w_n occurs in the document collection. This is the relative frequency of the word w_n in the document collection. Now, given the documents as sequences

2. Latent Variable Models

of randomly drawn words we can calculate the probability of any document as

$$p(\mathbf{d}) = \frac{\sum_{w \in \mathbb{V}} n_{d,w}}{\prod_{w \in \mathbb{V}} n_{d,w}!} \prod_{w_i \in \mathbb{V}} p(w_i)^{n_{d,w}},$$

for $n_{d,w}$ the number of occurrences of word w in the sequence of words from document d and \mathbb{V} the vocabulary of the corpus. This is the definition of the multinomial distribution. For example, given a corpus of two documents: “ $a b a c$ ” and “ $a a c$ ”, represented as the sequences: $\mathbf{d}_1 = (a, b, a, c)$ and $\mathbf{d}_2 = (a, a, c)$. Using the Uni-gram LM, we get the following word probabilities:

$$p(a) = \frac{4}{7} \qquad p(b) = \frac{1}{7} \qquad p(c) = \frac{2}{7}. \qquad (2.1)$$

Now, we can estimate the probabilities of the documents. For document \mathbf{d}_1 , for example, we have:

$$p(\mathbf{d}_1) = \frac{4}{2} p(a)^2 p(b) p(c).$$

Different Language Models are also possible. For instance in a tri-gram LM, we estimate conditional distributions $p(w_n | w_{n-2}, w_{n-1})$ additional to the marginal distributions $p(w_n)$ of word w_n in the sequence $\mathbf{d} = (w_1, \dots, w_N)$. The conditional probabilities can be estimated using the frequencies of the tri-grams in the corpus. Under this LM, the probability of a sequence of words is

$$p(\mathbf{d}) \propto p(w_1) p(w_2) \prod_{n=3}^N p(w_n | w_{n-1}, w_{n-2}).$$

Here, we assume independence of the words in the document given two previous words. The last equation is noted proportional since we might have to normalize the left hand side to get a proper probability distribution.

In contrast to the VSM, the MM allows a probabilistic interpretation of the documents and words in the text collection: A document is represented as sequences of random draws of words. The content of the document stems from a combination of word probabilities which are specified by a LM.

The VSM and the MM represent documents (the words) as variables: in the VSM we have deterministic variables, in the MM random variables. The documents from a given text collection in the corresponding representation are realizations of these variables. The realizations of the variables stem from a process that generates the words in each document. In the VSM this process is deterministic, in the MM this process is probabilistic. Deterministic means that each observed document will always have the same Word-Vector representation. The probabilistic process on the other hand results in variable representations as sequences of words for a document.

The generation process for the Word-Vectors of the documents in the VSM is:

1. For each document d :
 - a) For each word w_i in \mathbb{V} :



Figure 2.1.: Graphical representation of N independent variables w in the Plate notation. Left: notation of N independence variables as shaded rectangular nodes, right: summarized notation of N independent variable as N times a single variable. For example, the VSM with binary occurrences can be graphically represented in such a way.

- i. Draw $\mathbf{w}_{di} = a_{id}$

The value a_{id} of the variable \mathbf{w}_{di} depends on the individual representation as Bag-of-Words (Occurrences, TF, TF-IDF). Using pure occurrences for instance, the generation process builds the Word-Vectors as linear combination of the standard basis vectors \mathbf{w}^i in \mathbb{R}^V as described above:

$$\mathbf{w}_d = \sum_{i=1}^V a_{id} \mathbf{w}^i,$$

with $a_{id} \in \{0, 1\}$ indicating the presence (as 1) of word w_i in document d . Here, we assume total independence of the words in the document. This means, the components in the Word-Vectors have no information about possible correlations.

The generation process in the MM on the other hand generates for each document d the sequence (w_1, \dots, w_N) of words via:

1. For each document d :
 - a) For each word w_n in d :
 - i. Draw $w_n \sim p(w)$

The word probabilities $p(w)$ depend on the used LM such that $\mathbf{d} = (w_1, \dots, w_N) \sim p(\mathbf{d})$ for the joint probability $p(\mathbf{d})$. Using a uni-gram LM for example, the generation process builds the sequences as random draws from

$$p(\mathbf{d}) = \prod_{n=1}^N p(w_n).$$

As in the last example, we assume that each word is generated independently of all other words in the document.

Depending on the used Word-Vectors, respectively LM, the generation process imposes dependences among the variables for the realizations. The previous examples showed generation processes under fully independence assumptions among the words. The process (deterministic or random) produces the realizations of the variables w_i independent on all other realizations w_j for $j \neq i$. In Figures 2.1 and 2.2, this is graphically visualized by the so called Plate diagram.

2. Latent Variable Models



Figure 2.2.: Graphical representation of N independent random variables w . in the Plate notation Left: notation of N independence variables as shaded circular nodes, right: summarized notation of N independent variable as N times a single variable. For example, the MM with uni-gram LM can be graphically represented in such a way.

The Plate diagram or Plate notation formalizes a graphical model with (random) variables, dependencies and repetition as a graph. Although this notation is mainly used for probabilistic models, we also use this for the deterministic VSM (here we have only non-random variables). Given a document, the contained words are visualized as observed variables w_i . In the VSM, we show the variables as shaded rectangular nodes. In the MM, we show the random variables as shaded circular nodes. Dependence between the variables is shown by edges. For fully independent models, no edges are given - seen on the left of the figure. For simplification, equivalent variables are summarized as block with their multiplicity - seen on the right of the figures.

The full independence assumptions on the variables ignore possible correlations between the words (the realizations). Assuming full dependence between the variables on the other hand can include such correlation information in the generation process of the Word-Vectors, respectively the word sequences. As shown in Figure 2.3, assuming each variable depends on each other makes the model more complicated. On the other hand, this assumptions might result in better representations of the documents. A fully dependent VSM uses for example term frequency values in the Word-Vectors. The frequency value of a word depends on how many times this word is among the realizations of the variables and the number of variables in the Word-Vector. Given for instance a document containing the word *president* several times, will have a larger value in the Word-Vectors component that corresponds this word. A possible fully dependent MM on the other hand assumes that the realizations stem from conditional probabilities $p(w_i|w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_N)$. For instance, the occurrence of the word *States* as realization of variable w_i might depend on the occurrence of the word *United* as realization of variable w_{i-1} . Hence, $p(\text{States}|\text{United}) \geq p(\text{States}|w)$ for any other word w .

In real world corpora, we do not know the true generation process. We only know, that the observations are either vectors in \mathbb{R}^V for the VSM or random sequences drawn for multinomial distributions for the MM. The simplest way to describe the process of generating the realizations of the variables is to assume the fully independent model. In the VSM with occurrence counts in the Word-Vectors for instance, we can write each Word-Vector \mathbf{w}_d as sparse combination of the standard unit vectors \mathbf{w}^i in \mathbb{R}^V such that

$$\mathbf{w}_d = \sum_{i=1}^V a_{id} \mathbf{w}^i,$$

with $a_{id} \in \{0, 1\}$. In the MM on the other hand, the fully independent model assumes that the

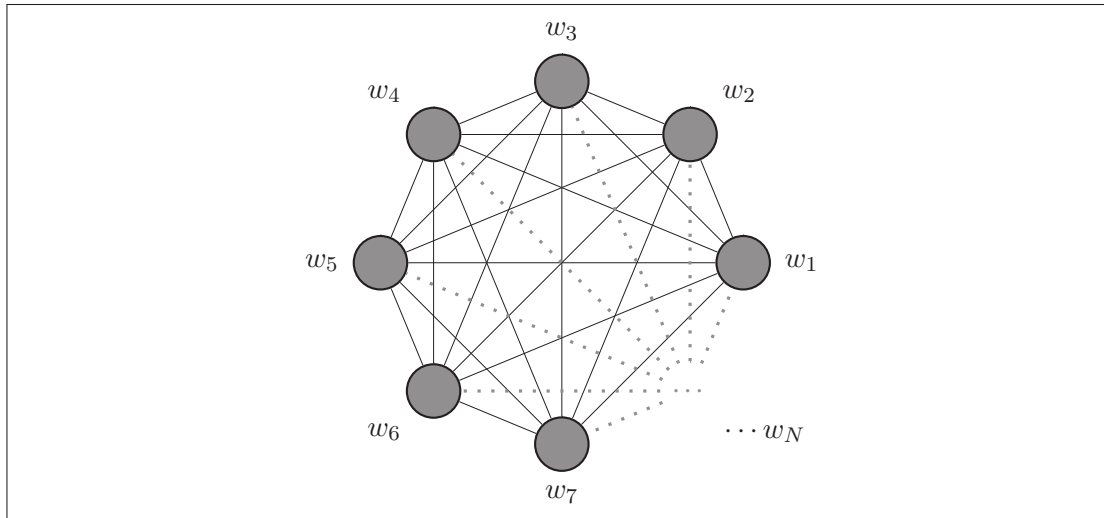


Figure 2.3.: Graphical representation of N fully dependent random variables w by a fully connected graph in the Plate notation. The dependence is visualized by edges. The VSM with TF Word-Vectors and the MM with N-gram LM follow this dependence assumptions.

sequence \mathbf{d} of words is distributed via

$$p(\mathbf{d}) = \prod_{n=1}^N p(w_n) = \prod_{w_i \in \mathbb{V}} p(w_i)^{n_{di}},$$

for n_{di} the number of occurrences of word w_i in document d . These descriptions have little expressiveness, since we model the data as high dimensional with many variables and no information of correlations between the variables. Hence, the fully independent model is only a weak approximation of the true generation process. A fully depend model as described above on the other hand could remedy the missing correlations between the variables, but at the expense of that we need all variables to describe the process.

In order to reduce the number of variables needed to describe the process of generating Word-Vectors for the VSM and random sequences for the MM, we use latent variable models. This models a relation between the observed variables and a number of latent variables. Here, latent means we do not observe these variables, we have to extract them from the data. The observed variables can be described by a small number of these latent variables. Using latent variables, we approximate the process of generating the realizations by modeling the observed variables as conditional independent given latent variables.

Conditional independent variables are independent of each other, given additional variables. We assume that the observed variables can be fully described given the latent variables. In Figure 2.4, we show the graphical representation of observed random variables w , that are conditional independent given latent (unobserved) random variables t . Such an approach models

2. Latent Variable Models

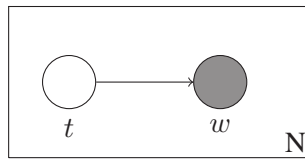


Figure 2.4.: Graphical representation of N conditionally independent random variables w , given the unobserved random variable t in the Plate notation. The conditional dependence is visualized by an arrow.

the correlation between the variables over common latent variables. Given a small number of latent variables, this description is more expressive as the fully independent approach and less complex than the fully dependent approach. The latent variable models we consider are all of this type: We have a number of observed variables w with a large number of possible realizations - the words. These variables depend on latent variables t with a smaller number of realizations $t \in \{1, \dots, T\}$. This reduces the dimensionality of the data from V to T .

Using the latent variable model to approximate the generation process of the words represented as Word-Vectors or random sequence, we can describe the process with fewer variables (latent variables).

Before we go into the details of different latent variable methods, we will discuss the interpretation of these latent variables in terms of the document context. Based on the different representations of the documents, we can assume that the document content does not only stem from the basis in the VSM or the marginal word probabilities in the MM. We assume that the content stems from a number of latent informations which originates from a combination of words.

This assumption originates from different works from linguistics and information retrieval. First, we assume that the content of each document can be described by contexts. The context as for instance discussed by [DG92], identifies the focus of a text. It can be used to distinguish documents and words by different contexts. Identifying the contexts by NLP methods can be done by analysis of the documents in the different representations in the VSM or in the MM. The context is collected in the Word-Vectors for the VSM and in the random sequences in the MM. The Word-Vectors and random sequences from the documents describe the context by co-location information of other words. Hence, the context is described by the words. This assumption goes back to research for word senses in linguistics in the 60s. John Rupert Firth, for instance, investigated this in [Fir57]. He is known for the sentence:

“You shall know a word by the company it keeps.”

In [RG65] Rubenstein and Goodenough investigated this assumption on synonymy. They studied how similarity between contexts correlates with similarities of meanings of words. Further, Wittgenstein already stated [Wit53]:

“The meaning of a word is its use in language.”

Relation	Description
synonym	different words with same meaning
polysemy	same word with different sense
homonymy	similar pronounced words with different meaning
hyponym	subordinate word
hypernym	superordinate word
antonym	oppositional word

Table 2.1.: Word sets based on relations among words.

A study on how contexts given by co-occurring words can describe language is given by Zellig Harris in [Har81]. He introduced the distribution of a language element (a word for example) as sum of the environments. Each such environment consists of all co-occurring elements of a given element. Regularly co-occurring elements in these environments define structure - distributional structure. The environments can be seen as the contexts described above. Further, elements (words) can be grouped such that given the group, the elements have similar distributional structure.

As pointed out by Koll in [Kol79], relations between words can be more easily described by underlying concepts. The underlying concepts summarize the meaning of the words. The term meaning in this context means the intent behind words or documents. For words, this can be different senses, synonyms, hyponyms, hypernyms or antonyms. For documents, this can be a subject in the text collection. The meaning summarizes the content of the documents and the words as intention of the texts. In linguistics the study of meanings is called semantics.

We distinguish two major fields for semantics in linguistics [Par95]: Lexical semantics and compositional semantics. Lexical semantics concentrates on words and word senses. Words can be grouped into classes based on relations among them. The word senses and these classes are parts of dictionaries. The most prominent word classes and the relations are reported in Table 2.1.

Compositional semantics on the other hand concentrates on meanings based on parts in the texts and combinations of these text parts. Complex meanings can be extracted from a context by small text fragments and their combinations in context. While traditional compositional semantics considers also syntax to identify parts in texts that convey the meaning, we concentrate only on words as part of a text and co-occurrences in a given context.

Classical approaches from linguistics extract such meaning by hand. For example by substitution, words are grouped into semantically related sets if they they can be exchanged in example sentences. Glinz [Grö73] proposes a test by substituting words to find semantically related words like synonyms. If we can simply replace a given word in a sentence with another word without changing the meaning of the sentence, we expect these two words to be synonym. In contrast to the classical approach that replaces words in context (several sentences including a sentence with the word of interest) and counts exact matches, the work in this thesis estimates statistics of similar contexts. Based on the distribution of co-occurring words possible substitutions are automatically extracted.

Considering the distributional structure as introduced by Harris, the words with similar distri-

2. Latent Variable Models

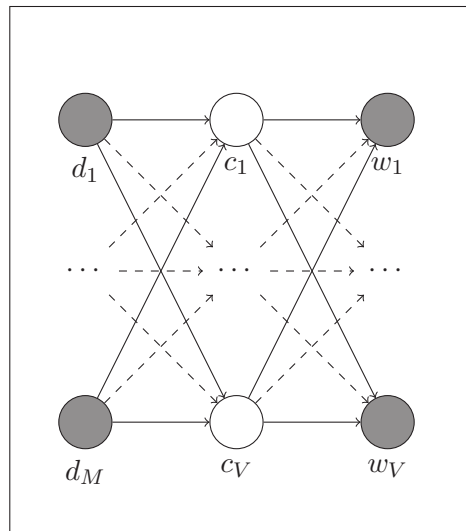


Figure 2.5.: Mapping of documents to concepts and concepts to words. The observed words in the documents depend on concepts that are induced by the documents themselves.

butional structure can be grouped to concepts. Each concept can be described by the words and the distributional structure. Hence, the words and documents build semantic groups associated with the concepts. Each concept can be connected to a meaning. The latent variable methods extract mappings of words and documents to these concepts to infer meaning. By these maps, the words and documents are related to the concepts with different strength based on statistics. The latent variables that are used to approximate the generation process of the documents are associated with the concepts. In Figure 2.5 we illustrate the mapping of words and documents to concepts by Plate notation using random variables.

2.1. Overview

In this section, an overview on the existing latent variables methods for document collections is given. We distinguish between latent factor models in the Vector Space Model and latent topic models in the Multinomial Model. The notation latent factors originates from factorization. The latent factor models factorize the Term-Document Matrix in a product of a document-concept matrix L and concept-word matrix R :

$$X \approx LR.$$

The notation of a topic originates from [PTRV98]. The authors introduce a topic as probability distribution of terms (words) that reflect the word distribution with respect to a certain subject. The latent topic model models the joint probability of words and documents as mixture over latent concepts c (later called topics t):

	VSM	MM
Observations	Word-Vectors	Sequence of words
Process	$\mathbf{w}_d = \sum a_{id} \mathbf{w}^i$	$(w_1, \dots, w_N) \sim p(\mathbf{d})$
Latent Variables	Vectors \mathbf{v}^i	Random variables t_1, \dots, t_T
Approximated Process	$\mathbf{w}_d \approx \sum_i \omega_{id} \mathbf{v}^i$	$p(w_1, \dots, w_N) \approx \prod_n \sum_t p(w_n t)p(t)$

Table 2.2.: Approximation of the generation process of the documents in the different representations.

$$p(w) \approx \sum_c p(w|c)p(c).$$

This notation is short for $\sum_{i=1}^T p(w|c=i)p(c=i)$.

Both models assume that the generation of the documents in their corresponding representation can be approximated by combinations of latent variables. For example, given a document as Word-Vector $\mathbf{w}_d \in \mathbb{R}^V$, we assume that

$$\mathbf{w}_d \approx \sum_{i=0}^T \omega_{id} \mathbf{v}^i,$$

for the T factors \mathbf{v}^i . Hence, we approximate the Word-Vector by a linear combination of the column vectors from the concept-word matrix with weights ω_{id} .

For a document d as a sequence $\mathbf{d} = (w_1, \dots, w_N)$ of word-tokens (or simply tokens) w_n drawn from a probability distribution $p(w)$, we assume

$$p(\mathbf{d}) \approx \prod_n \sum_t p(w_n|t)p(t),$$

for the topics t . Hence, the words conditionally depend on the topics with probability $p(w_n|t)$. In Table 2.2, we summarize the generation process for the VSM and the MM and the corresponding approximations by latent variables.

The latent variables are used to extract certain meanings or groupings of the words and the documents. Based on co-occurring words, either in common Word-Vectors or in sequences of tokens from the documents, these groupings induce meanings. For example consider the following text: *"It is raining cats and dogs. In such weather I rather stay at home and watch TV."* By the model of distributional structure, we assume that the meaning of each word in this text is determined by the other words present in the text. For instance, we assume that the word "cats" together with "and", "dogs" and "raining" is meant in a metaphorical way rather than as an animal. This gets more support considering that the word "weather" is also present in the text.

A document can have several such meanings in certain amounts. While at first sight the text above might be assigned to the meaning of weather rather than animals, the concept animal is present. Intuitively speaking, we assume that the meaning of the above text is a combination of weather and animals.

2. Latent Variable Models

This combination of meanings is expressed as combination of latent variables. In the VSM, latent factor models express the Word-Vectors as weighted sum of vectors (latent factors) that summarize certain meanings in the vector space spanned by the words (respectively the documents). In the MM, latent topic models express the document and word probabilities as weighted combination of (multinomial) probabilities conditioned on latent topics that summarize certain meanings.

2.2. Factor Models

Factor models assume that the Word-Vectors, representing the documents, can be expressed as a combination of certain vectors, also called factors. These factors can be used as summarization of the documents or as low-dimensional representation of the Word-Vectors. The underlying assumption for factor models in language processing is that the context in which a word appears determines its meaning. The mathematical background of factor models is linear algebra. The documents are represented as vectors in a certain space (usually a Euclidean or a Hilbert space). Within this space a subspace that contains certain (or all) moments of the data is extracted. A basis of this subspace can be used as factors. In the next subsections we explain the most prominent factor models. First, we explain a factor model that assumes that the documents are represented as Word-Vectors in a Euclidean space. Second, we discuss how factor models can be estimated when we assume that the documents are represented as high (possibly infinite) dimensional vectors in a Hilbert space.

2.2.1. Latent Semantic Analysis

Latent Semantic Analysis (LSA) as described by Landauer et al. in [LD97] and Deerwester et al. in [DDF⁺90] extracts usage patterns in documents by grouping words into latent dimensions in the vector space. LSA assumes that words that appear in common text documents are also semantically related. In the Vector Space Model, the Word-Vectors represent the co-occurrences of words in a certain document. The Term-Document Matrix is factorized by a Singular Value Decomposition (SVD) [GVL96] to extract low dimensional subspace in the space spanned by the documents and in the space spanned by the terms. We factorize the Term-Document Matrix such that

$$X = LER,$$

for the concept-term matrix R that consists of the right singular vectors of X , the document-concept matrix L that consists of the left singular vectors of X and E the diagonal matrix of the singular values of X . The singular vectors define a basis in the space of the Word-Vectors and in the space of the Document-Vectors. The singular values sum up the lengths of the projections of the Word-Vectors onto the (right) singular vector. The larger this value, the more variance of the Word-Vectors lies in this dimension (the dimension that is spanned by the singular vector). Sorting the singular values, we get a ranking of the singular vectors that span the subspace that contains most of the information of the Word-Vectors. The diagonal matrix E_T contains the T largest singular values and the approximation

$$X \approx LE_T R,$$

$$\begin{aligned}
 X &= LER \\
 &= \underbrace{\begin{pmatrix} l_1 & & & & l_N \\ & \dots & & & \\ & & & & \end{pmatrix}}_{\text{span}(X)} \underbrace{\begin{pmatrix} e_1 & & & & \\ & \ddots & & & \\ & & & & e_N \end{pmatrix}}_{\text{Concepts}} \underbrace{\begin{pmatrix} \text{---} & r'_1 \\ & \\ & \\ & \\ \text{---} & r'_V \end{pmatrix}}_{\text{span}(X')}
 \end{aligned}$$

Figure 2.6.: Illustration of the matrix decomposition in LSA by an SVD. The term-document matrix X is factorized into the matrix L that contains the left singular vectors (interpreted as concept distributions in each document), the diagonal matrix E that contains the singular values (interpreted as intensity of each concept) and the matrix R that contains the right singular vectors (interpreted as word distributions for the concepts). This results in the decomposition $X = LER$. In LSA, we use the largest singular values to identify those singular values that span the subspace that contains most of the information from X . This subspace is called the semantic subspace and the largest singular values with the corresponding singular vectors are associated with concepts present in the document represented as Word-Vectors in X .

is the best rank T approximation of the Term-Document Matrix X . This means $LE_T R$ is a projection of the Word-Vectors and the document-vectors onto a T -dimensional subspace. The projected Word-Vectors and document-vectors in this subspace have the lowest reconstruction error among all possible subspaces. Hence,

$$\|X - LE_T R\|_2^2$$

is minimized. In Figure 2.6, we schematically illustrate LSA by a Singular Value Decomposition. The geometric interpretation of LSA as extracting a basis spanned by the singular vectors is sketched in Figure 2.7.

This means, the T left singular vectors that correspond to the largest singular values span the T dimensional subspace in the document space and the right singular values in the term space that contain most of the variance of the Word-Vectors. The value of the components of the right singular vectors multiplied by the corresponding singular values indicate the variance of the terms in a certain direction of the subspace. The space spanned by the first T singular vectors is called the concept-space. The components of the singular vectors describe the importance of the words for a concept. The biggest of these values can be interpreted as the terms most important

2. Latent Variable Models

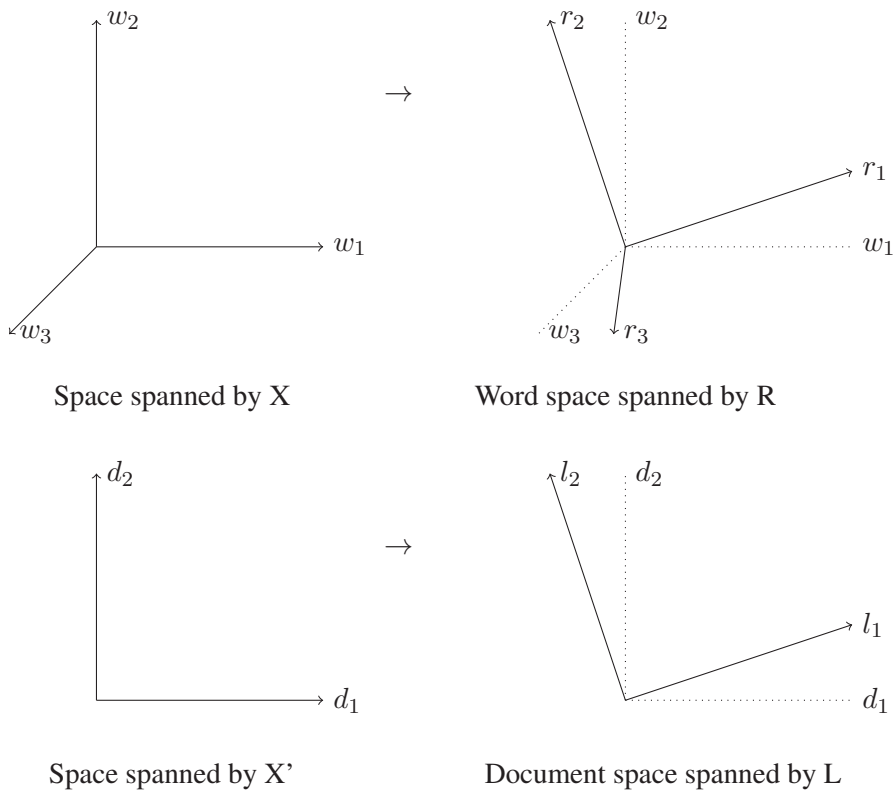


Figure 2.7.: Illustration of the decomposition of the matrix X . Given two documents d_1, d_2 with three words w_1, w_2, w_3 the Word-Vectors in X span a space as depicted on the left. SVD extracts a new basis in the space spanned by the words and a new basis in the space spanned by the documents illustrated on the right. Using the first T left and right singular vectors results in two subspaces in the words and documents space. This subspace can be interpreted as spanned by concepts.

in a certain usage pattern. Each singular vectors is associated with a concept and we use them as factors to approximate the Word-Vectors.

In LSA we assume that the Word-Vectors \mathbf{w}_d can be expressed as linear combinations of latent vectors \mathbf{v}^i plus a normally distributed error term ϵ_d :

$$\mathbf{w}_d = \sum_i^T \omega_i \mathbf{v}^i + \epsilon_d$$

with $\omega_i \in \mathbb{R}$ and $\mathbf{v}^i \in \mathbb{R}^V$. Using SVD to find the basis vectors, we can truncate the right singular vectors with small singular values and still keep a large amount of the information from the data to define the factors as the right singular vectors.

There are several algorithms to perform an SVD. The simplest method is the power method, see [KW92]. This method, as described in Algorithm 1, can be used to extract the left and right

Algorithm 1 Power method for SVD.

```

function POWERMETHOD( $X$ )
  for  $i = 1 : T$  do
     $\mathbf{w} = urand$  // random initialization
    repeat
       $\mathbf{w} := X'X\mathbf{w}$ 
    until convergence
     $\mathbf{r}_i = \mathbf{w} / \|\mathbf{w}\|$ 
     $e_i = \|X\mathbf{r}_i\|$ 
     $\mathbf{l}_i = X\mathbf{r}_i / e_i$ 
     $X = X'X(I - \mathbf{r}_i\mathbf{r}_i')$ 
  end for
  return  $[\mathbf{l}_i]_{i=1 \dots T}, [e_i]_{i=1 \dots T}, [\mathbf{r}_i]_{i=1 \dots T}$ 
end function

```

singular vectors. The method uses the property that the sequence $(X'X)^i \mathbf{w}_d$ converges to the first (largest to the corresponding singular value) right singular value of X . To see this, we write $\mathbf{w}_d = \sum_i \omega_i \mathbf{r}_i$ as linear combination of the eigenvectors \mathbf{r}_i of $(X'X)^1$, where X' is the transpose². Since \mathbf{r}_i span a basis in \mathbb{R}^V , we can define \mathbf{w}_d in this way. Multiplying \mathbf{w}_d by $(X'X)^j$ results in:

$$\begin{aligned}
 (X'X)^j \mathbf{w}_d &= (X'X)^j \sum_i \omega_i \mathbf{r}_i & (2.2) \\
 &= \sum_i \omega_i (X'X)^j \mathbf{r}_i \\
 &= \sum_i \omega_i e_i^{2j} \mathbf{r}_i \\
 &= e_1^{2j} c_1 \mathbf{r}_1 + e_1^{2j} \sum_{i>1} \omega_i \left(\frac{e_i}{e_1}\right)^{2j} \mathbf{r}_i.
 \end{aligned}$$

The right hand side of Equation 2.2 converges for $j \rightarrow \infty$ to $e_1^{2j} c_1 \mathbf{r}_1$ under the assumption that $e_1 > e_j$. Normalizing this results in the right singular vector \mathbf{r}_1 . The corresponding singular value is the length of the projection onto \mathbf{r}_i , i.e. $e_i = \|X\mathbf{r}_i\|$. Since $X'\mathbf{l}_1 = e_1 \mathbf{r}_1$ by the definition of singular vectors, we get the left singular vectors by: $\mathbf{l}_1 = X\mathbf{r}_1 / e_1$. After the extraction of the first left and right singular vectors, we deflate the corresponding eigenspace from $X'X$, by projecting $X'X$ onto its orthogonal subspace: $X = X'X(I - \mathbf{r}_i\mathbf{r}_i')$. Next, we extract the eigenvectors in the deflated space as before and continue until we have extracted all T vectors.

In LSA, the left singular vectors and the right singular vectors are used as a semantic grouping of words and documents based on the magnitude of the length of the projections onto them. For

¹Note that the eigenvectors of $(X'X)$ are the same as the right singular vectors of X . To see this we write $A = LER$. We get $A'A = (LER)'LER = R'EL'LER = R'E^2R$

²Throughout this work, we use the symbol ' for the transpose.

Algorithm 2 Partial Least Squares to extract the latent factors.

```

function GETCOMPONENT( $X, \mathbf{y}, T$ )
  for  $i = 1 : T$  do
     $\mathbf{u} = urand$  // random initialization
    repeat
       $\mathbf{w} = X' \mathbf{u}$ 
       $\mathbf{v}_i = X \mathbf{w}, \mathbf{v}_i = \mathbf{v}_i / \|\mathbf{v}_i\|$ 
       $c = \mathbf{y}' \mathbf{v}_i$ 
       $\mathbf{u}^i = \mathbf{y} c, \mathbf{u}^i = \mathbf{u}^i / \|\mathbf{u}^i\|$ 
    until convergence
     $X = X - \mathbf{v}^i \mathbf{v}^{i'}$ 
     $y = y - \mathbf{v}^i \mathbf{v}^{i'}$ 
  end for
  return  $V = [\mathbf{v}^i]_{i=1 \dots T}$  and  $U = [\mathbf{u}^i]_{i=1 \dots T}$ 
end function

```

example, the right singular vector \mathbf{r} contains at some components (indices of the vector) large absolute values. The words that span the corresponding dimensions in the vector space can be grouped together. This group is interpreted as prominent in this latent dimension and treated as semantically connected words. They belong to the same concept and have similar meanings.

Next, we explain a similar method when additional labels of documents can be explicitly taken into account. While LSA finds only semantic subspaces describing the documents, we can also look for subspaces that reflect the given label information about the documents.

2.2.2. Partial Least Squares

Partial Least Squares (PLS) is a method that finds low dimensional subspaces, which maximally align with label information. Given documents as Word-Vectors and labels of the documents, PLS finds low dimensional Word-Vector representations that are the optimal covariates for a linear regressor to predict the labels. These labels can be time stamps indicating the date of publication of the documents or information about the content like sentiment information or a classification of the tone used in the document. Algorithm 2 describes the steps of PLS for a given Term-Document Matrix X of Word-Vectors and a label vector \mathbf{y} as described by Rosipal and Trejo in [RT02]. The algorithm successively extracts latent factors as linear combinations of the input Word-Vectors. These components are removed from the Word-Vectors by deflating the term-document matrix by $X - \mathbf{v} \mathbf{v}' X$. Deflating means we remove the dimension spanned by \mathbf{v} by projecting all Word-Vectors onto its orthogonal complements. This process is repeated until we have found T components. This method is analogue the power method used for SVD.

The result of the algorithm are so called loadings vectors. These vectors can be seen analogous to the singular vectors extracted by SVD. Each loadings vector is a low dimensional representation of a corresponding Word-Vector. These loadings can be used to estimate the amount of rotation the words in the vector space experience when mapped onto the vectors - similar to the interpretation of the singular vectors in LSA.

The extracted loading vectors can be easily used to predict new unlabeled documents. Note that PLS is a linear regression model. The label is simply modeled as regression: $\mathbf{y} = X\boldsymbol{\omega} + \mathbf{r}$ for the term document matrix X , the regression coefficients $\boldsymbol{\omega}$ mapping onto the latent factors and a residual vector \mathbf{r} . For $V = [\mathbf{v}^1, \dots, \mathbf{v}^T]$ and $U = [\mathbf{u}^1, \dots, \mathbf{u}^T]$ from PLS via Algorithm 2, we can estimate the coefficients by

$$\boldsymbol{\omega} = X'U(V'XX'U)^{-1}V'\mathbf{y}. \quad (2.3)$$

A new document represented as Word-Vector \mathbf{w}_d is assigned label $\text{sign}(\mathbf{w}'_d\boldsymbol{\omega})$ for binomial labels like sentiments, respectively $\mathbf{w}'_d\boldsymbol{\omega}$ for numeric labels like time spans.

The interpretation of the latent factors, respectively loading vectors is not simple. One aspect for the importance of a word for one latent factor is the value of the corresponding component in the loading vector. The amount of the j th component of \mathbf{v}^i tells how much weight the corresponding word has to predict the label when we project it onto the latent factor \mathbf{v}^i . In order to better interpret the importance of some words for the latent factors we can rotate the loading vectors such that the variance in these vectors is maximized. Intuitively, we want loadings that have few large components and near zero components elsewhere. The method Varimax Rotation [Kai58] does exactly this. This method rotates the coordinate system spanned by the latent factors such the loadings in the new coordinate system have maximum variance in their components.

For LSA and PLS is difficult since components can be arbitrary, positive or negative. It would be more intuitive if we could model (the positive) word presence (or frequency) as combination of certain amounts (positive) of latent factors. This results in a simpler interpretation of factors.

2.2.3. Non-negative Matrix Factorization

While LSA factorizes the Term-Document Matrix via an SVD, Non-negative Matrix Factorization (NNMF) factorizes the Term-Document Matrix in the product of two non-negative matrices. Consequently, we assume that each document can be expressed as positive mixture of positive factors. This enables a more intuitive interpretation of the factors compared to LSA. Formally, we model each document d as

$$\mathbf{w}_d = V\boldsymbol{\omega}_d + \boldsymbol{\epsilon}_d,$$

such that $\boldsymbol{\omega}_d \in \mathbb{R}_+^V$ for all documents d , $V = [\mathbf{v}^1, \dots, \mathbf{v}^T]$ with $\mathbf{v}^i \in \mathbb{R}_+^V$ and with no assumption on $\boldsymbol{\epsilon}_d$. Under this model we want to find two non-negative matrices $W = [\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_M] \geq 0$ and $V \geq 0$, such that

$$X \approx WV.$$

In [LS01], Lee and Seung introduce algorithms for this factorization. The authors propose to extract the corresponding matrices by minimizing the l_2 matrix norm

$$\begin{aligned} \text{dist}(X, WV) &:= \|X - WV\|_2^2 \\ &= \sum_{i,j} (X_{i,j} - (WV)_{i,j})^2, \quad W \geq 0, V \geq 0, \end{aligned} \quad (2.4)$$

2. Latent Variable Models

Algorithm 3 Nonnegative Matrix Factorization.

```

function GETNNMF( $X, k$ )
   $W = urand, V = urand$  // random initialization
   $k = 0$ 
  repeat
     $W^{k+1} = \arg \min_{W \geq 0} \text{dist}(X, WV^k)$ 
     $V^{k+1} = \arg \min_{V \geq 0} \text{dist}(X, W^{k+1}V)$ 
     $k = k + 1$ 
  until convergence
end function

```

respectively the KL-Divergence

$$\text{dist}(X, WV) := \sum_{i,j} (X_{i,j} \log \frac{X_{i,j}}{(WV)_{i,j}} - X_{i,j} + (WV)_{i,j}), W \geq 0, V \geq 0.$$

Since both distance functions are non-convex in both arguments, but convex keeping one argument fix, Lee and Seung propose to solve

$$\min_{W \geq 0, V \geq 0} \text{dist}(X, WV) \quad (2.5)$$

in an alternating fashion. To solve the optimization we alternate between minimizing the distance function with respect to W keeping V fix and with respect to V keeping W fix. Since the distance function is convex in a single argument this will converge, but likely to a local minimum. In Algorithm 3 we summarize these steps.

Depending on the concrete distance function, we iteratively (over k) solve the two optimization problems

$$\arg \min_{W \geq 0} \text{dist}(X, WV^k), \quad (2.6)$$

respectively

$$\arg \min_{V \geq 0} \text{dist}(X, W^{k+1}V) \quad (2.7)$$

analytically or by gradient descent. If we use, for example, the distance function from Equation 2.4 we get the following two gradients:

$$\begin{aligned} \nabla_W \text{dist}(X, WV) &= \frac{1}{2} W'(WV - X) \\ \nabla_V \text{dist}(X, WV) &= \frac{1}{2} V'(WV - X) \end{aligned}$$

As discussed by Lin in [Lin07], projected gradient methods can be used to solve the optimization problems in Equations 2.6 and 2.7 by Stochastic Gradient Descent [Bot98] and Projected

Algorithm 4 Projected Gradient Method.

```

function GETARGMIN( $X, W, V^k$ )
   $W = urand$  // random initialization such that  $W \geq 0$ 
  repeat
     $W = P[W - \lambda_k \nabla_W \text{dist}(X, WV^k)]$ 
     $k = k + 1$ 
  until convergence
end function

```

Gradients Methods. The details are given in Algorithm 4. The step size λ_k can be estimated by line search methods and the projection P is defined as

$$P[X] = \begin{cases} X, & \text{if } X \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Similar to LSA and PLS, the column vectors in V and the row vectors in W can be interpreted as semantic grouping of the words and the documents.

An interesting extension of NNMF was proposed by Liu and Wu in [LW10]. The authors include document labels as hard constraints in the optimization for NNMF. The constraints force the Word-Vectors from documents with the same label to be mapped into the same low-dimensional latent feature representation.

The previous methods performed a factorization of the documents in a Euclidean space using Word-Vectors. Besides this BoW representation of the documents, different more structured representations are also possible. In the next subsection, we describe a non-linear factorization methods that can be performed in arbitrary Hilbert spaces.

2.2.4. Kernel Principal Component Analysis

Kernel methods accomplish to apply linear methods on non-linear representations of data. Any kernel method uses a map from a compact input space - we focus on \mathbb{R}^V - into a so called Reproducing Kernel Hilbert Space (RKHS). In this space, linear methods are applied to the mapped elements like Linear Regressions or Support Vector Machines. The RKHS H is a space of functions that allows point evaluations by inner products:

$$f(\mathbf{w}_d)(\mathbf{w}) = \langle \phi(\mathbf{w}), \phi(\mathbf{w}_d) \rangle,$$

where $\phi(\mathbf{w}_d)$ is a function in H and $f(\mathbf{w}_d)(\mathbf{w})$ is the function value for the Word-Vector \mathbf{w} for the function $f(\mathbf{w}_d)$ indexed at \mathbf{w}_d . For example, using the Bag-of-Words as features, we have $(\phi(\mathbf{w}_d))_i = a_{di}$ a V -dimensional vector having at component i the frequency of word w_i for instance.

For the mapping ϕ from above, K_ϕ is the integral operator for a probability distribution P on the input space X . It is defined as

$$K_\phi(f)(\mathbf{w}_d) = \int f(\mathbf{w}) \langle \phi(\mathbf{w}), \phi(\mathbf{w}_d) \rangle dP(\mathbf{w}). \quad (2.8)$$

2. Latent Variable Models

For this integral operator, we denote $\langle \phi(\mathbf{w}), \phi(\mathbf{w}_d) \rangle = k(\mathbf{w}, \mathbf{w}_d)$ with a kernel $k(\mathbf{w}, \mathbf{w}_d)$. By Mercer Theorem [Mer09] there is a one to one correspondence of the above defined RKHS and the integral operator via the kernel $k(x, y)$. This correspondence is given by the expansion

$$k(\mathbf{w}, \mathbf{w}_d) = \sum_{i=1}^{\infty} \phi_i(\mathbf{w})\phi_j(\mathbf{w}_d)$$

for $\{\phi_i\}$ an orthonormal basis in the RKHS.

Now, the covariance operator C on a Hilbert space H is defined as $E[Z \times Z^*]$ the outer product of a random element $Z \in H$ with its adjoint Z^* . This is analogue to the covariance of centered random elements in \mathbb{R}^V where we have $C = E[XX']$ for the Term-Document Matrix X . The empirical covariance is estimated via $\hat{C} = \frac{1}{M} \sum \phi(\mathbf{w}_{d_i})\phi(\mathbf{w}_{d_j})'$ for a centered sample $\{\phi(\mathbf{w}_{d_1}), \dots, \phi(\mathbf{w}_{d_M})\}$ with \mathbf{w}_{d_i} drawn from distribution P . Consequently the kernel matrix approximates the covariance operator: $K \sim C$.

Schölkopf et al. proposed in [SSM99] to perform Principal Component Analysis (PCA) [Hot33] in a kernel defined RKHS based on the eigenfunctions and eigenvalues of the covariance operator C to extract low-dimensional approximations of the mapped data.

Analogue to SVD, we use PCA to extract orthogonal vectors that span a subspace in the data space that contains most of the variance. In contrast to standard SVD, in PCA we use the covariance matrix $C = XX'$ to extract the vectors. The connection between SVD and PCA is straight. While SVD uses the data matrix to perform the following factorization:

$$X = UEV'.$$

PCA performs this factorization:

$$XX' = UE^2U'.$$

We can use the power method for SVD to get the factorization since

$$XX' = UEV'(UEV') = UEV'VEU = UE^2U.$$

Similar to LSA, we extract eigenvectors such that the mapped data $\phi(\mathbf{w}_d)$ can be expressed as linear combination of these vectors. Kernel Principal Component Analysis (kPCA) extracts an orthogonal basis, also called principal components, in a kernel induced RKHS. Projecting the data onto the subspace spanned by the first T components captures most of the variance among the data compared to all other possible subspaces where the data lies in. The T components are exactly the eigenfunctions corresponding to the largest T eigenvalues of the covariance operator of the kernel.

An eigenvalue decomposition on C results in a set of eigenvalues $\{\lambda_i\}$ and eigenvectors $\{\mathbf{v}^i\}$ such that $\lambda_i \mathbf{v}^i = C\mathbf{v}^i$. A projection of a sample \mathbf{w}_d in the RKHS onto $U = \{\mathbf{v}^i\}$ is done by

$$P_U(\phi(\mathbf{w}_d)) = (\langle \mathbf{v}^1, \phi(\mathbf{w}_d) \rangle, \dots, \langle \mathbf{v}^T, \phi(\mathbf{w}_d) \rangle) \in U.$$

Since, the \mathbf{v}^i lie in the span of the $\{\phi(\mathbf{w}_{d_i})\}$, each component is given by $\mathbf{v}^i = \sum_j \omega_{j,i} \phi(\mathbf{w}_{d_j})$. This results in the projection:

$$P_U(\phi(\mathbf{w}_d)) = \left(\sum_j \omega_{j,1} \langle \phi(\mathbf{w}_{d_i}), \phi(\mathbf{w}_d) \rangle, \dots, \sum_j \omega_{j,T} \langle \phi(\mathbf{w}_{d_i}), \phi(\mathbf{w}_d) \rangle \right) \in U.$$

Algorithm 5 Kernel Principal Component Analysis.

Center kernel matrix \bar{K}
 Perform eigenvalue decomposition: $[V, \Lambda] = \text{eig}(\bar{K})$
 Calculate kernel matrix K^P of the mapped data samples onto the subspace

Algorithm 6 Kernel Partial Least Squares to extract the latent factors.

```

function GETCOMPONENT( $K, Y$ )
  ...
   $\mathbf{u} = \text{urand}$  // random initialization
  repeat
     $\mathbf{v} = K\mathbf{u}, \mathbf{v} = \mathbf{v}/\|\mathbf{v}\|$ 
     $c = \mathbf{y}'\mathbf{v}$ 
     $\mathbf{u} = \mathbf{y}c, \mathbf{u} = \mathbf{u}/\|\mathbf{u}\|$ 
  until convergence
   $K = (I - \mathbf{v}\mathbf{v}')K(I - \mathbf{v}\mathbf{v}')$ 
   $Y = (I - \mathbf{v}\mathbf{v}')Y(I - \mathbf{v}\mathbf{v}')$ 
   $\mathbf{v}^i = \mathbf{v}$ 
   $\mathbf{u}^i = \mathbf{u}$ 
   $i = i + 1$ 
  ...
  return  $[\mathbf{v}^i]_{i=1\dots T}$ 
end function

```

The eigenvalues can be calculated by: $\omega_{i,j} = (\frac{1}{\sqrt{\lambda_i}}\mathbf{v}^i)_j$.

The steps of kernel PCA are summarized in Algorithm 5 as described by Shawe-Taylor and Cristianini in [STC04].

Similar to PLS, we can also use document labels to find subspaces in the kernel defined RKHS that align to these labels. Compared the PLS, kernel Partial Least Squares (kPLS) can also use structured, high or even infinite dimensional labels for the documents.

2.2.5. Kernel Partial Least Squares

Similar to kPCA, kPLS performs PLS in a kernel defined Reproducing Kernel Hilbert Space (RKHS). From the definition of PLS, we see that computing a component \mathbf{v} is done by $\mathbf{v} = XX'\mathbf{u}$. The matrix XX' is the empirical covariance matrix between the Word-Vectors in X . This matrix is the approximated true covariance matrix for random Word-Vectors drawn from the same distributions as the Word-Vectors. The idea now is to apply kernel methods for the extractions of the latent factors.

The algorithm of kPLS is analogue to PLS. In Algorithm 6 we shortly show the differences compared to the standard PLS. The only differences are that we directly calculate \mathbf{v} as $K\mathbf{u}$ and that the projection onto the orthogonal complement respectively the deflation of \mathbf{v} is done by $(I - \mathbf{v}\mathbf{v}')K(I - \mathbf{v}\mathbf{v}')$.

Like PLS, kPLS can be used as regression to predict the labels for unlabeled documents. The

2. Latent Variable Models

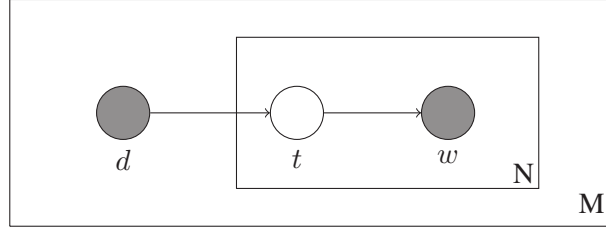


Figure 2.8.: pLSA represented as graphical model in the Plate notation. For M observed documents, each contained (observed) word depends on a latent variable t .

regression model is : $\mathbf{y} = \Phi\boldsymbol{\beta} + \mathbf{r}$ and the coefficients $\boldsymbol{\beta}$ can be estimated by

$$\boldsymbol{\beta} = \Phi'U(V'KU)^{-1}V'\mathbf{y},$$

with $V = [\mathbf{v}^1, \dots, \mathbf{v}^T]$ and $U = [\mathbf{u}^1, \dots, \mathbf{u}^T]$.

The interpretation of the latent factors \mathbf{v}^i is more difficult than before. Compared to LSA that finds subspaces in the space spanned by the words, kPCA finds subspaces that are spanned by large (possibly infinite) dimensional vectors. Hence, the factors \mathbf{v}^i are linear combinations of possible infinite dimensional Hilbert space elements. In order to interpret them, we can only investigate the documents that are mapped the closest to the one dimensional subspace spanned by each \mathbf{v}^i . The idea is that these documents contain the (possible not countable) structures that are important for the corresponding factors. This renders the interpretation of the factors into a Pre-Image problem as described by Gökhan et al. in [BWS04].

2.3. Topic Models

Topic models are statistical models that extract semantics in text corpora based on co-occurrence statistics. For these models, we assume the MM such that the words in the documents are drawn from multinomial distributions. The most prominent latent topic models are the probabilistic Latent Semantic Analysis and Latent Dirichlet Allocation. Both models are mixture models [MB88] that model the joint probability of words as linear combinations of conditional distribution of the latent topics.

The Aspect Model, as for instance used by Hofmann in [HPJ99], models the observed words in the documents dependent on unobserved latent variables. These random variables are the aspects, respectively the latent topics. Further, the words in the documents are conditionally independent given latent random variables and the latent variables are independent given a document. In the literature there is the notation of aspects synonym to topics. We use the notation of topics throughout this thesis. Both topics and aspects stand for a concept hidden in the documents.

2.3.1. Probabilistic Latent Semantic Analysis

One of the first models for co-occurrence data by an Aspect Model is probabilistic LSA (pLSA). It can be seen as a probabilistic version of Latent Semantic Analysis. As introduced by Hofmann in [Hof99], pLSA models the probability of the words in the documents as mixture over latent topics t . Hence, we assume that the probability of a word w in document d can be expressed as

$$p(w|d) = \sum_t p(w|t)p(t|d).$$

This notation is short for $\sum_{i=1}^T p(w|t=i)p(t=i|d)$. The overall joint probability is given by

$$\begin{aligned} p(d, w) &= p(d)p(w|d) \\ &= p(d) \sum_t p(w|t)p(t|d) \\ &= p(d) \sum_t p(w|t) \frac{p(t)p(d|t)}{p(d)} \\ &= \sum_t p(w|t)p(t)p(d|t), \end{aligned}$$

for $p(t|d) = \frac{p(t)p(d|t)}{p(d)}$ by the Bayes rule [KSO87]. The graphical representation of this joint probability is given in Figure 2.8 and the generative process³ for the words in a document can be summarized by

1. For each document d :
 - a) For each word w_n in document d :
 - i. Draw $t_n \sim p(t|d)$
 - ii. Draw $w_n \sim p(w|t_n)$

To find the conditional probabilities of the documents given a latent topic and the words given a latent topic, an Expectation Maximization (EM) algorithm [DLR77] is used. An EM algorithm iterates between an E-step that estimates a distribution empirically and an M-step that finds parameters that maximize the likelihood of the distribution. In the E-step we estimate the posterior distribution:

$$p(t|d, w) \propto p(t)p(d|t)p(w|t).$$

In the M-step we maximize the likelihood with respect to the parameters: $p(t)$, $p(d|t)$ and $p(w|t)$. The likelihood L , respectively the log-likelihood, for documents d as sequences $\mathbf{d} = (w_1, \dots, w_{N_d})$ is

³For the probabilistic models we use the term *generative process* instead of generation process that is used for non-probabilistic models as the factor model and the probabilistic models as the topic models.

2. Latent Variable Models

$$\begin{aligned}
 L &= \prod_d \prod_n^{N_d} p(d, w_n) \\
 &= \prod_d \prod_{w \in \mathbb{V}} \prod_n I[w = w_n] p(d, w) \\
 &= \prod_d \prod_w p(d, w)^{n(w, d)} \\
 \Rightarrow \log L &= \sum_d \sum_w n(w, d) \log p(d, w),
 \end{aligned}$$

for $n(w, d)$ the number of occurrences of word w in document d .

Setting the partial derivatives with respect to $p(w|t)$, $(d|t)$ and $p(t)$ to zero, we end up with the following update rules that maximize $\log L$:

$$\begin{aligned}
 p(w|t) &\propto \sum_d n(d, w) p(t|d, w) \\
 p(d|t) &\propto \sum_w n(d, w) p(t|d, w) \\
 p(t) &= \frac{\sum_d \sum_w n(d, w) p(t|d, w)}{\sum_d \sum_w n(d, w)}.
 \end{aligned}$$

The EM-algorithm alternates between the E-step and the M-step until convergence.

There is also a geometric interpretation of the pLSA analogue to LSA. While LSA extracts factors that span subspaces in the space spanned by the Word-Vectors, pLSA extracts probability distributions that span simplices. A simplex is the geometric object

$$\mathbb{S}^V = \{(p_1, \dots, p_V) \mid \sum_i p_i = 1\}$$

and can be interpreted as the set containing all multinomial distributions. Hence, the probability distribution $p(w|d)$ for each document lies in a probability simplex \mathbb{S}^V . The topics span a sub-simplex in \mathbb{S}^V by the topic-word distributions $p(w|t)$ such that $\sum_i p(w|t_i) = 1$. The probability distribution $p(t|d)$ is the projection of the probability distribution $p(w|d)$ onto this sub-simplex. In Figure 2.9, we illustrate this geometric interpretation.

There is an interesting connection between pLSA and NNMF: Both methods optimize the same objective but with different algorithms. Using the tact, that we can write the maximization of $\log L$ as minimization, Ding et al. in [DLP08] performed the following reformulations:

$$\begin{aligned}
 \arg \max \log L &= \arg \min -\log L \\
 \Rightarrow -\log L &= \sum_d \sum_w n(w, d) \frac{1}{\log p(d, w)}.
 \end{aligned}$$

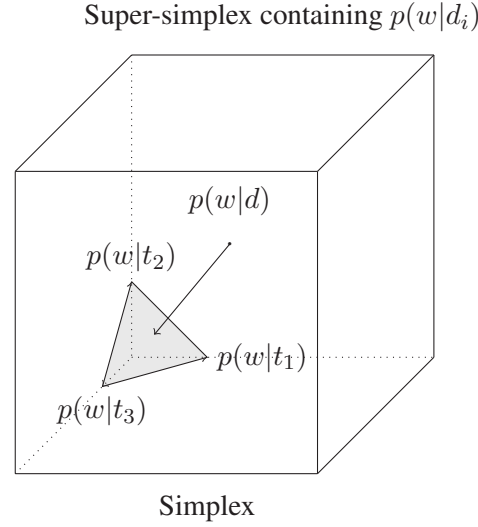


Figure 2.9.: Illustration of probabilistic LSA as decomposition of probability distributions. In the super-simplex containing all word-document probabilities $p(w|d)$, a sub-simplex is extracted that contains the word-topic distributions $p(w|t)$. The word-topic distribution $p(w|t)$ for document d is the projection of the word-document probabilities $p(w|d)$ onto this sub-simplex.

Scaling the left hand side of Equation 2.9 by $S = \sum_d \sum_w n(d, w)$ and adding the constant

$$\sum_d \sum_w \frac{n(d, w)}{S} \log \frac{n(d, w)}{S}$$

results in the equivalent formalization

$$\sum_d \sum_w \frac{n(w, d)}{S} \log \frac{n(w, d)}{p(d, w)}. \quad (2.9)$$

Since $p(w, d)$ is a probability we have $\sum_d \sum_w p(d, w) = 1$ and $\sum_d \sum_w \frac{n(d, w)}{S} = 1$, adding $\sum_d \sum_w p(d, w) - \frac{n(d, w)}{S}$ to the previous equation results in the same optimum. All these reformulations lead to the following optimization problem:

$$\min \sum_d \sum_w \log \frac{n(w, d)}{S} \frac{n(w, d)}{p(d, w)} + p(d, w) - \frac{n(d, w)}{S}. \quad (2.10)$$

If we set $(p(d, w))_{d, w} = X = WH^S$ we get the objective of NMF as proposed by Lee and Suang in [LS01].

2.3.2. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) as proposed by Blei et al. [BNJ03] is a generative probabilistic topic model similar to Probabilistic Latent Semantic Analysis. The difference to the previous model is the additional assumption of Dirichlet priors on the document-topic and the topic-word distributions.

Given a corpus C of M documents each represented by sequences of words $\mathbf{d} = (w_1, \dots, w_N)$, LDA models the generative process of generating documents as random draws over random mixtures of latent topics. We briefly summarize the generative process of documents as the following:

1. For each topic t :
 - a) Draw $\beta_t \sim \text{Dir}(\eta)$
2. For each document $d \in C$:
 - a) Draw $\theta_d \sim \text{Dir}(\alpha)$
 - b) For each word w_n in document d :
 - i. Draw $t_n \sim \text{Mult}(\theta_d)$
 - ii. Draw $w_n \sim \text{Mult}(\beta_{t_n})$

First, we draw for each topic t the word probabilities β_t for each word in the corpus. Next, for each document we draw a T -dimensional Dirichlet distributed random vector θ_d . Then, for each token in the document d we draw a topic t_n from a multinomial distribution parametrized with θ_d and a word w_n from a multinomial distribution parametrized with β_{t_n} . In the original approach by Blei et al., β_t does not have a Dirichlet prior $\text{Dir}(\eta)$. This becomes important for sampling based approaches for LDA and for possible extensions with different (more complicated) priors.

In the literature there are conceptually two major approaches to estimate an LDA topic model. First, variational inference can be used to approximate the posterior distribution of the latent variables by a simpler variational distribution. Second, Gibbs sampling defines a sequence of random draws that converges to a sequence of topic assignments that follows the joint distribution of the topic model.

Variational Inference for LDA

In Variational Inference complex posterior distributions are approximated by simple distributions that are close in terms of a divergence measure like the KL-divergence⁴. A general introduction into Variational Inference methods can be found in [JGJS99] by Jordan et al.

For LDA, the posterior distribution is given by

$$p(\theta, \beta, \mathbf{t} | \mathbf{d}, \alpha, \eta) = \frac{p(\theta, \beta, \mathbf{t}, \mathbf{d} | \alpha, \eta)}{p(\mathbf{d} | \alpha, \eta)} \quad (2.11)$$

⁴The KL-divergence measures the distance between two probability distribution p and q . It is calculated as :
 $KL(p||q) = \int_x p(x) \log \frac{p(x)}{q(x)}$.

for $\mathbf{t} = (t_1, \dots, t_N)$ given topic assignments for each token and with the joint probability in the nominator

$$p(\theta, \beta, \mathbf{t}, \mathbf{d}|\alpha, \eta) = p(\theta|\alpha) \prod_{t=1}^T p(\beta_t|\eta) \prod_{n=1}^N p(t_n|\theta)p(w_n|t_n, \beta)$$

and the marginal distribution of a document in the denominator

$$p(\mathbf{d}|\alpha, \eta) = \int p(\theta|\alpha) \left(\prod_{t=1}^T p(\beta_t|\eta) \prod_{n=1}^N \sum_{t'=1}^T p(t'|\theta)p(w_n|t', \beta_{t'}) \right) d\theta d\beta.$$

The random variables θ and β are Dirichlet distributed and lie in the $(T-1)$ -simplex with probability density

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{t=1}^T \alpha_t)}{\prod_{t=1}^T \Gamma(\alpha_t)} \theta_1^{\alpha_1-1} \dots \theta_T^{\alpha_T-1}.$$

Since the posterior distribution in Equation 2.11 is intractable, a so called variational distribution $q(\theta, \beta, \mathbf{t}|\gamma, \lambda, \phi)$ with variational parameters γ , λ and ϕ approximates the posterior distribution $p(\theta, \beta, \mathbf{t}|\mathbf{w}, \alpha, \eta)$. This variational distribution shall have minimum KL-divergence to p . The KL-divergence $D(q||p)$ is minimized when we maximize the term

$$L(\gamma, \lambda, \phi; \alpha, \eta) = \mathbb{E}_q[\log p(\theta, \beta, \mathbf{t}|\alpha, \eta)] - \mathbb{E}_q[\log q(\theta, \beta, \mathbf{t}|\gamma, \lambda, \phi)]. \quad (2.12)$$

This is justified by the following inequality based on Jensen's inequality [Jen06]:

$$\log p(\mathbf{d}|\alpha, \eta) = \log \int \sum_{\mathbf{t}} p(\theta, \beta, \mathbf{t}, \mathbf{d}|\alpha, \eta) d\theta d\beta \quad (2.13)$$

$$= \log \int \sum_{\mathbf{t}} \frac{p(\theta, \beta, \mathbf{t}, \mathbf{d}|\alpha, \eta) q(\theta, \beta, \mathbf{t})}{q(\theta, \beta, \mathbf{t})} d\theta d\beta \quad (2.14)$$

$$\geq \int \sum_{\mathbf{t}} q(\theta, \beta, \mathbf{t}) \log \frac{p(\theta, \beta, \mathbf{t}, \mathbf{d}|\alpha, \eta)}{q(\theta, \beta, \mathbf{t})} d\theta d\beta \quad (2.15)$$

$$\begin{aligned} &= \int \sum_{\mathbf{t}} (q(\theta, \beta, \mathbf{t}) \log p(\theta, \beta, \mathbf{t}, \mathbf{d}|\alpha, \eta) \\ &\quad - \log q(\theta, \beta, \mathbf{t})) d\theta d\beta \\ &= \int \sum_{\mathbf{t}} \log p(\theta, \beta, \mathbf{t}, \mathbf{d}|\alpha, \eta) q(\theta, \beta, \mathbf{t}) d\theta d\beta \quad (2.16) \\ &\quad - \int \sum_{\mathbf{t}} \log q(\theta, \beta, \mathbf{t}) q(\theta, \beta, \mathbf{t}) d\theta d\beta \end{aligned}$$

$$= E_{q(\theta, \beta, \mathbf{t})} \log p(\theta, \beta, \mathbf{t}, \mathbf{d}|\alpha, \eta) - E_{q(\theta, \beta, \mathbf{t})} \log q(\theta, \beta, \mathbf{t}).$$

Equation 2.13 writes $p(\mathbf{d})$ as marginal distribution over the random variable θ , β and \mathbf{t} . In Equation 2.14 the inner addend is expanded by $\frac{q(\theta, \beta, \mathbf{t})}{q(\theta, \beta, \mathbf{t})}$. Since this is like multiplying with 1,

2. Latent Variable Models

we did not change the equation. Next in Equation 2.15, Jensen's inequality is applied. This is possible since the logarithm is a concave function, respectively the negative logarithm is a convex function. Further, the right hand side of Equation 2.14 is the logarithm of the expectation of $\frac{p(\theta, \beta, \mathbf{t}, \mathbf{d}|\alpha, \eta)}{q(\theta, \beta, \mathbf{t})}$ under $q(\theta, \beta, \mathbf{t})$. Consequently, we can apply Jensen's inequality. In the remaining equations the terms are rearranged such that in the end we get a bound for an arbitrary variational distribution q .

If we add the KL-divergence of p and q on the right hand side of Equation 2.13, the inequality becomes an equality. The KL-divergence of p and q is

$$\begin{aligned} D(q(\theta, \beta, \mathbf{t}|\gamma, \lambda, \phi) \| p(\theta, \beta, \mathbf{t}|\mathbf{d}, \alpha, \eta)) &= \int q(\theta, \beta, \mathbf{t}|\gamma, \lambda, \phi) \log \frac{q(\theta, \beta, \mathbf{t}|\gamma, \lambda, \phi)}{p(\theta, \beta, \mathbf{t}|\mathbf{d}, \alpha, \eta)} \\ &= \int q(\theta, \beta, \mathbf{t}|\gamma, \lambda, \phi) \log q(\theta, \beta, \mathbf{t}|\gamma, \lambda, \phi) \\ &\quad - \int q(\theta, \beta, \mathbf{t}|\gamma, \lambda, \phi) \log p(\theta, \beta, \mathbf{t}|\mathbf{d}, \alpha, \eta). \end{aligned}$$

The first term of the right hand side of the last equation can be rewritten as

$$\begin{aligned} E_{q(\theta, \beta, \mathbf{t})} \log p(\theta, \beta, \mathbf{t}, \mathbf{d}|\alpha, \eta) &= E_{q(\theta, \beta, \mathbf{t})} \log p(\theta, \beta, \mathbf{t}|\mathbf{d}, \alpha, \eta) p(\mathbf{d}|\alpha, \eta) \\ &= E_{q(\theta, \beta, \mathbf{t})} \log p(\theta, \beta, \mathbf{t}|\mathbf{d}, \alpha, \eta) + \log p(\mathbf{d}|\alpha, \eta) \\ &= E_{q(\theta, \beta, \mathbf{t})} \log p(\theta, \beta, \mathbf{t}|\mathbf{d}, \alpha, \eta) + E_{q(\theta, \beta, \mathbf{t})} \log p(\mathbf{d}|\alpha, \eta) \\ &= E_{q(\theta, \beta, \mathbf{t})} \log p(\theta, \beta, \mathbf{t}|\mathbf{d}, \alpha, \eta) + \log p(\mathbf{d}|\alpha, \eta). \end{aligned}$$

Here, we use the fact that $p(\theta, \beta, \mathbf{t}, \mathbf{d}|\alpha, \eta) = p(\theta, \beta, \mathbf{t}|\mathbf{d}, \alpha, \eta)p(\mathbf{d}|\alpha, \eta)$ and $\mathbb{E}_q c = c$ if c does not depend on q . Finally, we can reformulate the lower bound to

$$L(\gamma, \lambda, \phi; \alpha, \eta) + D(q(\theta, \beta, \mathbf{t}|\gamma, \lambda, \phi) \| p(\theta, \beta, \mathbf{t}|\mathbf{d}, \alpha, \eta)) = \log p(\mathbf{d}|\alpha, \eta). \quad (2.17)$$

Since $\log p(\mathbf{d}|\alpha, \eta)$ does not depend on the variational parameters γ , λ and ϕ , it can be seen as constant. Consequently, minimizing the KL-divergence is the same as maximizing $L(\gamma, \lambda, \phi; \alpha, \eta)$. Now, we only need to specify an appropriate variational distribution with variational parameters. Based on the original graphical model for LDA (see Figure 6.2), a simple graphical model for the variational distribution is derived. The variational distribution with varying variational parameters builds a family of functions that shall be used as lower bounds for the posterior distribution $p(\mathbf{d}|\alpha, \eta)$. To gain tight bounds, Blei et al. [BNJ03] propose to derive a simpler graphical model by removing some edges and nodes from the original model. This results in the following variational distribution:

$$q(\beta, \theta, \mathbf{t}|\gamma, \phi) = q(\theta, \gamma) \prod_{t=1}^T q(\beta_t|\lambda) \prod_{n=1}^N q(t_n|\phi_n). \quad (2.18)$$

Now everything is at hand to optimize L from Equation 2.12 by the following variational EM algorithm:

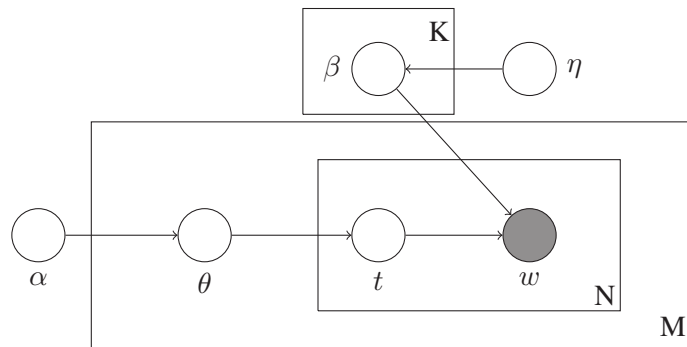


Figure 2.10.: LDA represented as graphical model in the Plate notation. For M documents, we draw a topic distribution θ with Dirichlet prior $\text{Dir}(\alpha)$. For each of the N tokens in a document, we draw a topic t from θ . Given the topic and the topic-word distribution β , we draw the words.

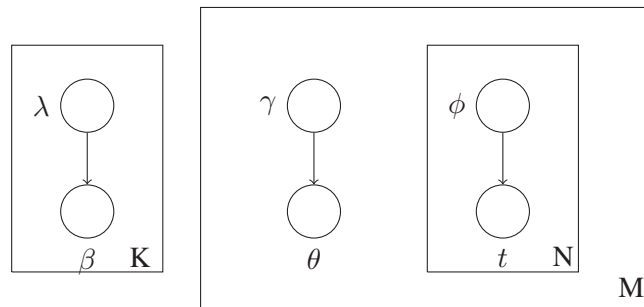


Figure 2.11.: Variational distribution for LDA as graphical model in the Plate notation.

1. (E) Find optimal variational parameters γ_d, ϕ_d for each document d and λ_t for each topic t by maximizing the likelihood from Equation 2.12.
2. (M) Find maximum likelihood estimate from the model parameters α, η with sufficient statistics estimated under the posterior from the E-step.

The single steps for the optimization in the E-step and the parameter estimation in the M-step can be found in the original LDA paper by Blei et al. In the following, we summarize the most important equations for the steps:

To calculate L in Equation 2.12, we need several expectations. Here, we must take care that we always take the expectation with respect to the variational distribution q . By the definitions of the posterior distribution p and the variational distribution q , we can rewrite L to

2. Latent Variable Models

$$\begin{aligned}
L(\gamma, \lambda, \phi; \alpha, \eta) &= \mathbb{E}_q[\log p(\theta|\alpha)] + \sum_t \mathbb{E}_q[\log p(\beta_t|\eta)] \\
&+ \sum_n \mathbb{E}_q[\log p(t_n|\theta)] + \sum_n \mathbb{E}_q[\log p(w_n|t_n, \beta)] \\
&- \mathbb{E}_q[\log (q(\theta|\gamma))] - \sum_t \mathbb{E}_q[\log (q(\beta_t|\lambda))] \\
&- \sum_n \mathbb{E}_q[\log (q(t_n|\phi_n))].
\end{aligned} \tag{2.19}$$

We summarize the most important terms in the following. The expectation of the log-probability of a Dirichlet distributed random variable θ is

$$\begin{aligned}
\mathbb{E}_q[\log p(\theta|\alpha)] &= \mathbb{E}_q[\log \exp(\sum_t (\alpha_t - 1) \log \theta_t + \log \Gamma(\sum_t \alpha_t) - \sum_t \log \Gamma(\alpha_t))] \\
&= \sum_t (\alpha_t - 1) \mathbb{E}_q[\log \theta_t] + \log \Gamma(\sum_t \alpha_t) - \sum_t \log \Gamma(\alpha_t) \\
&= \sum_t (\alpha_t - 1) (\Psi(\alpha_t) - \Psi(\sum_{t'} \alpha_{t'})) + \log \Gamma(\sum_t \alpha_t) - \sum_{t'} \log \Gamma(\alpha_{t'}),
\end{aligned}$$

and the expectation of the log-probability of a word is

$$\begin{aligned}
\mathbb{E}_q[\log p(\mathbf{d}|t, \beta)] &= \int \sum_{t,n} \log \beta_{t,w_n} q(t) q(\beta) d\beta \\
&= \int \sum_{t,n} \log \beta_{t,w_n} \phi_{t,n} q(\beta) d\beta \\
&= \sum_{t,n} \phi_{t,n} \int \log \beta_{t,w_n} q(\beta) d\beta \\
&= \sum_{t,n} \phi_{t,n} \mathbb{E}_q[\log \beta_{t,w_n}].
\end{aligned}$$

These are all expectations of the factors of the joint probability in L . Finally, we need the expectation of the variational distribution $q(\mathbf{t})$:

$$\begin{aligned}
\mathbb{E}_q[\log(q(\mathbf{t}|\phi))] &= \mathbb{E}_q[\log(\prod_{n=1}^N q(t_n|\phi))] \\
&= \mathbb{E}_q[\sum_{n=1}^N \log(q(t_n|\phi))] \\
&= \sum_t \sum_{n=1}^N \log(q(t_n|\phi))q(t_n|\phi) \\
&= \sum_t \sum_{n=1}^N \log(\phi_{t,n})\phi_{t,n}.
\end{aligned}$$

Remembering that $q(t|\phi) = \phi_{t,\cdot}$, the expectation of the logarithm of a Dirichlet distributed random variable θ is

$$E_{q(\theta|\alpha)}[\log \theta_i] = \Psi(\alpha_i) - \Psi(\sum_j \alpha_j)$$

and inserting the corresponding expectations in the lower bound results in the equation as shown in Figure 2.12. This is the final lower bound that is optimized. Minimizing the bound, the updates of the M-step are summarized in the equations in Figure 2.13.

Online LDA by Stochastic Variational Inference

Online LDA, as introduced by Hoffmann et al. [HBB10], uses Stochastic Gradient Descent (SGD) to find the optimal variational distribution (the parameters). To account for the non-Euclidean geometry of the parameters, a Riemann metric between probability distributions is used. So called natural gradients are used for the SGD. The natural gradient is especially easy to compute for probability distributions of the exponential family.

During the SGD two parameter sets are distinguished, the local parameter depending on a document d and the global parameters independent of certain documents. The local parameters are γ_d and ϕ_{dn} , the global parameters are λ_{tv} . The local parameters are estimated for each document as in standard LDA. The global parameters are updated based on the current topic distributions and the estimates from the last iteration j :

$$\begin{aligned}
\hat{\lambda}_{t,v} &= \eta_{t,v} + D \sum_{n=1}^N \phi_{d,n}^t w_{d,n} \\
\lambda_{t,v}^{j+1} &= (1 - \rho_j) \lambda_{t,v}^j + \rho_j \hat{\lambda}_{t,v}.
\end{aligned}$$

The stochastic variational variance can be summarized as in Algorithm 7.

2. Latent Variable Models

$$\begin{aligned}
& L(\gamma, \lambda, \phi; \alpha, \eta, d) \\
&= \log \Gamma\left(\sum_{j=1}^T \alpha_{dj}\right) - \sum_{j=1}^T \log \Gamma(\alpha_{dj}) + \sum_{i=1}^T ((\alpha_{di} - 1)\Psi(\gamma_{di}) - \Psi\left(\sum_{j=1}^T \gamma_{dj}\right)) \\
&+ \log \Gamma\left(\sum_{j=1,v=1}^{T,V} \eta_{jv}\right) - \sum_{j=1,v=1}^{T,V} \log \Gamma(\eta_{jv}) + \sum_{i=1,v=1}^{T,V} ((\eta_{iv} - 1)\Psi(\lambda_{iv}) - \Psi\left(\sum_{j=1,v=1}^{T,M} \lambda_{jv}\right)) \\
&+ \sum_{n=1,i=1}^{N,T} \phi_{ni} (\Psi(\gamma_{di}) - \Psi\left(\sum_{j=1}^T \gamma_{dj}\right)) \\
&+ \sum_{n=1,i=1,v=1}^{N,T,V} \phi_{ni} w_n^v (\Psi(\lambda_{i,v}) - \Psi\left(\sum_{j=1}^T \lambda_{j,v}\right)) \\
&- \log \Gamma\left(\sum_{j=1}^T \gamma_{dj}\right) - \sum_{j=1}^T \log \Gamma(\gamma_{dj}) + \sum_{i=1}^T ((\gamma_{di} - 1)\Psi(\gamma_{di}) - \Psi\left(\sum_{j=1}^T \gamma_{dj}\right)) \\
&- \log \Gamma\left(\sum_{j=1,v=1}^{T,V} \lambda_{jv}\right) - \sum_{j=1,v=1}^{T,V} \log \Gamma(\lambda_{jv}) + \sum_{i=1,v=1}^{T,V} ((\lambda_{iv} - 1)\Psi(\lambda_{iv}) - \Psi\left(\sum_{j=1,v=1}^{T,M} \lambda_{jv}\right)) \\
&- \sum_{n=1,i=1}^{N,T} \phi_{ni} \log \phi_{ni}
\end{aligned}$$

Figure 2.12.: Lower bound that is maximized in LDA.

$$\begin{aligned}
\phi_{dni} &\propto \beta_{iwn} \exp\left(\Psi(\gamma_{di}) - \Psi\left(\sum_{j=1}^T \gamma_{dj}\right)\right) \\
\gamma_{di} &= \alpha_{di} + \sum_{n=1}^N \phi_{dni} \\
\beta_{ij} &\propto \sum_{d=1,n=1}^{M,N} \phi_{dni} w_{dn}^j \\
\lambda_{ij} &= \eta_{ij} + \sum_{d=1,n=1}^{M,N} \phi_{dni} w_{dn}^j
\end{aligned}$$

Figure 2.13.: Updates for the parameters in LDA in the M-step.

Algorithm 7 Online algorithm for LDA

```

Initialize  $\lambda^0$ 
repeat
  Sample  $w_d$ 
  Initialize  $\gamma$ 
  repeat
     $\phi_{d,n}^t \propto \exp(E[\log \theta_{d,t}] + E[\log \beta_{t,w_n}])$ 
     $\gamma_{d,t} = \alpha_{d,t} + \sum_{n=1}^N \phi_{d,n}^t$ 
  until convergence
   $\hat{\lambda}_{t,v} = \eta_{t,v} + D \sum_{n=1}^N \phi_{d,n}^t w_{d,n}$ 
   $\lambda_{t,v}^{j+1} = (1 - \rho_j) \lambda_{t,v}^j + \rho_j \hat{\lambda}_{t,v}$ 
until done

```

Gibbs Sampling for LDA

Variational inference only approximates the true posterior distribution. The quality of this approximation highly depends on how good the variational distribution can approximate the true distribution. Another approach for inference with complicated posterior distributions are Markov Chain Monte Carlo (MCMC) methods, see [GS90] for an introduction. The idea of MCMC methods is to define a sequence of random draws such that after a number of such draws, these samples follow a certain distribution of interest. In the case of LDA, we want that the samples follow the joint distribution $p(t, w)$.

Using a Gibbs sampler (as MCMC method) [GG84] for example, we draw the topics directly from the topic distribution given only conditional distributions. In [GS04], Griffiths and Steyvers applied this on LDA for topic models. The idea is not to sample all topic assignments for given documents and words at once, but each at a time. Hence, beside the words, we also observe the topics $\mathbf{t}^{-i} = (t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n)$, only topic t_i remains an unobserved random variable. Iterative sampling one topic t_i for a word w_i given all other topic assignments as fixed, converges to a sequence of samples from the joint probability $p(t, w)$ of all topics and words. This is easy to show by the following equation:

$$\begin{aligned}
 p(t_i | \mathbf{t}^{-i}, w_i) &= \frac{p(\mathbf{t}_{+t_i}^{-i}, w_i)}{p(\mathbf{t}^{-i}, w_i)} \\
 &\propto p(\mathbf{t}, w_i),
 \end{aligned} \tag{2.20}$$

for $\mathbf{t}_{+t_i}^{-i} = (t_1, \dots, t_{i-1}, t_i, t_{i+1}, \dots, t_n)$. Since the $p(w_i, t)$ are independent for all words w_i , the Gibbs sampler results in the joint probability $p(t, \mathbf{d}) = \prod_n p(t, w_n)$.

To derive the corresponding joint probabilities from Equation 5.1, the random variables from $p(t, \mathbf{d})$ are integrated out. Then, we get for the probability of a sequence of topics t_n and a sequence of words w_n from a document d under the generative model of LDA:

2. Latent Variable Models

$$\begin{aligned}
p(\mathbf{t}, \mathbf{d}|\alpha, \eta) &= \int \int p(\mathbf{t}, \mathbf{d}, \theta, \beta|\alpha, \eta) d\theta d\beta & (2.21) \\
&= \int \int p(\theta|\alpha) p(\beta|\eta) p(\mathbf{t}|\theta) p(\mathbf{d}|\mathbf{t}, \beta) d\theta d\beta \\
&= \int p(\mathbf{t}|\theta) p(\theta|\alpha) d\theta \int p(\mathbf{d}|\mathbf{t}, \beta) p(\beta|\eta) d\beta.
\end{aligned}$$

Due to independence and the definition of the multinomial distribution, we have $p(\mathbf{t}|\theta) = \prod_n p(t_n|\theta) = \prod_n \theta_{t_n}$ and $p(\mathbf{d}|\mathbf{t}, \beta) = \prod_n \beta_{t_n, w_n}$. Finally, a random sequence $\mathbf{t} = (t_1, \dots, t_N)$ of topic assignments for a token sequence $\mathbf{d} = (w_1, \dots, w_N)$ from documents has probability $p(\mathbf{t}|\theta) = \prod_{t_n} \theta_{t_n}^{n_{d, t_n}}$. The sequences of tokens themselves have probability $p(\mathbf{d}|\mathbf{t}, \beta) = \prod_{w_n} \beta_{t_n, w_n}^{n_{t, w_n}}$. Since, we integrate whole random variables out, this sampling method is called collapsed Gibbs sampling.

We denote $n_{t, w}$ the number of times topic t has been assigned to word w , $n_{d, t}$ the number of times topic t has been assigned to any word in document d , further n_t the number of times topic t has been assigned to any word, V the number of words in the vocabulary from the document collection and d_n the number of tokens in document d (words with multiplicity).

Since the prior distributions $p(\theta|\alpha)$ and $p(\beta|\eta)$ are Dirichlet distributions which are conjugate to the multinomial distribution, the two terms on the right hand side of Equation 2.21 can be easily calculated.

Remember the definition of the Dirichlet distribution⁵, the two terms in Equation 2.21 can be reformulated as

$$\begin{aligned}
\int p(\mathbf{t}|\theta) p(\theta|\alpha) d\theta &= \prod_d \int \frac{1}{B(\alpha)} \prod_{t_n} \theta_{t_n}^{n_{d, t_n} + \alpha_{t_n} - 1} d\theta \\
&= \prod_d \frac{B(n_d + \alpha)}{B(\alpha)} \int \prod_{t_n} \frac{1}{B(n_d + \alpha)} \theta_{t_n}^{n_{d, t_n} + \alpha_{t_n} - 1} d\theta \\
&= \prod_d \frac{B(n_d + \alpha)}{B(\alpha)} \int \text{Dir}(n_d + \alpha) d\theta \\
&= \prod_d \frac{B(n_d + \alpha)}{B(\alpha)},
\end{aligned}$$

and

⁵ $\text{Dir}(\theta|\alpha) = \frac{1}{B(\alpha)} \prod_k \theta^{\alpha_k - 1}$ for the Beta function $B(\alpha) = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)}$

$$\begin{aligned}
\int p(\mathbf{d}|\mathbf{t}, \beta)p(\beta|\eta)d\beta &= \prod_{t_n} \int \frac{1}{B(\eta)} \prod_w \beta_{t_n, w}^{n_{t_n, w} + \eta_w - 1} d\beta \\
&= \prod_{t_n} \frac{B(n_{t_n} + \eta)}{B(\eta)} \int \text{Dir}(n_{t_n} + \eta) d\theta \\
&= \prod_{t_n} \frac{B(n_{t_n} + \eta)}{B(\eta)}.
\end{aligned}$$

Now, we can write the joint probability as

$$p(\mathbf{t}, \mathbf{d}|\alpha, \eta) = \prod_d \frac{B(n_d + \alpha)}{B(\alpha)} \prod_{t_n} \frac{B(n_{t_n} + \eta)}{B(\eta)}.$$

For the definition of the conditional distribution $p(t_i|\mathbf{t}^{-i})$ we use n_d^{-i} for the number times any topic has been assigned to any token in document d when we exclude the assignment t_i , hence $n_d^{-i} = n_d - 1$ and $n_{t,w}^{-i}$ for the number of times topic t has been assigned to word w when we exclude the assignment t_i , hence $n_{t,w}^{-i} = n_{t,w} - 1$. Further, we use the definition of the Beta function and the following equality: $\Gamma(x + 1) = x\Gamma(x)$.

Finally, we get the following conditional distribution for the Gibbs sampler:

$$\begin{aligned}
p(t_i|\mathbf{t}^{-i}, w) &= \frac{p(\mathbf{t}_{+t_i}^{-i}, w)}{p(\mathbf{t}^{-i}, w)} \tag{2.22} \\
&\propto \prod_d \frac{B(n_d + \alpha)}{B(n_d^{-i} + \alpha)} \prod_t \frac{B(n_t + \eta)}{B(n_t^{-i} + \eta)} \\
&\propto (n_{d,t}^{-i} + \alpha) \frac{n_{t,w}^{-i} + \eta_w}{\sum_{w'} n_{t,w'}^{-i} + \eta_{w'}}.
\end{aligned}$$

After a sufficient number of samples from the Gibbs sampler we get estimates of the word distributions for the topics and the topic distributions for the documents. Given the topic assignments $\{t_{d,w}\}$ for the words w in the documents d , we get

$$\beta_{w|t} = p(w|t) = \int p(w|\beta)p(\beta)d\beta = \frac{n_{w,t} + \eta_w}{n_t + \sum_{w'} \eta_{w'}}$$

and

$$\theta_{d|t} = p(t|d) = \int p(t|\theta)p(\theta)d\theta = \frac{n_{d,t} + \alpha_d}{n_d + \sum_{d'} \alpha_{d'}}.$$

Online LDA by Resampling Topics

Analogue to the online version of Variational Inference, Gibbs sampling strategies are also used in an online manner. In [YMM09], Yao et al. propose to use Gibbs sampling for online LDA. The authors test several strategies to sample topic assignments for new documents. Based on a converged run of the Gibbs sampler on a training set, the topic assignments so far are either kept fixed and the new documents are used one by one or all at once to sample the assignments for them, or all new documents and the old training documents are used together to re-sample the topic assignments. Further to these strategies, Canini et al. [CSG09] propose a different sampling strategy based on particle filters. They extend Gibbs sampling to perform several weighted samples. If the variance of the weights gets too big, they re-sample to adapt for possible changes in distribution.

Further Solutions for LDA

Variational inference (online or not) and Gibbs sampling are the most prominent solutions to estimate topics in LDA. Besides these methods two additional methods are commonly used: Belief Propagation and Expectation Propagation.

With Belief Propagation, the graphical model of LDA is interpreted as factor graph. The conditional probabilities $p(t_i | t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n, w)$ are interpreted as messages being passed in the factor graph, see [ZCL11] for more details. In Expectation Propagation on the other hand, we approximate $p(w|\theta) = \sum_t \theta_{d,t} \beta_{t,w}$ by $q(w|\theta) = s_w \prod_t \theta_{d,t}^{\beta_{t,w}}$. In an iterative manner, the influence of a word w is removed from q and the new s_w and β are estimated such that p and q have matching moments. An overview on Expectation Propagation for Generative Models can be found in [ML02].

Besides these different estimation methods for LDA, Variational Inference and Gibbs sampling have been optimized to reduce its complexity. In [NCL07], for example, Nallapati et al. propose a parallel version of the Variational Inference for LDA. The authors implement the E-step such that batches of documents are processed in parallel or even distributed. This is possible since the variational parameters γ and ϕ for each document are independent of the other documents. Further, in [NSWA08] Newman et al. introduce a hierarchical version of LDA with distributed θ and β across p processors or machines. The authors use Gibbs sampling to sample topic assignments for each processor, respectively machine, based on only local documents.

Teh et al. propose in [TNW07] a combination of Variational Inference and Gibbs sampling for LDA called collapsed variational inference. The authors show that this is achieved by jointly modeling θ and β in the variational distribution without further assumptions,

$$q_2(\theta, \beta, \mathbf{t}) = q_2(\theta, \beta | \mathbf{t}) \prod_n q(t_n | \phi_n),$$

as approximation of the true posterior p .

Considering all these different inference methods for LDA, Asuncion et al. investigate in [AWST09] their connections. The authors show that all method results in similar update

rules during the optimization of LDA. The difference is in different priors for the distributions. This once again shows the importance of meta parameters for the Dirichlet priors.

There is also an interesting connection between LDA and pLSA. As indicated in the beginning of this section, pLSA differs from LDA by the prior distributions. In [GK03] Girolami and Kaban showed that with uniform priors on θ and β , we get

$$\begin{aligned}
p(\theta, \beta, \mathbf{w}|\alpha, \eta) &= \prod_d p(\theta_d|\alpha) \prod_{t=1}^T p(\beta_t|\eta) \prod_{n=1}^N \sum_t p(t|\theta_d) p(w_n|t, \beta) \\
&\propto \prod_d \prod_{n=1}^{N_d} \sum_t p(t|\theta_d) p(w_n|t, \beta) \\
&= \prod_d \prod_w \left(\sum_t p(t|\theta_d) p(w|t, \beta) \right)^{n(d,w)} \\
&= \prod_d \prod_w p(w|\theta, \beta)^{n(d,w)} \\
\log p(\theta, \beta, \mathbf{w}|\alpha, \eta) &= \sum_d \sum_w n(d, w) \log p(w|\theta, \beta).
\end{aligned}$$

The last equation is the log posterior for LDA with uniform priors and is equivalent to the log posterior for pLSA.

Supervised Topic Models

Similar to PLS and kPLS that integrate document labels in the extraction of latent factors, topic models can also be augmented to integrate document labels. Supervised topic models integrate additional labels for each document such that the latent topics can be used to predict further unlabeled documents. As proposed by Blei and McCalliffe in [MB08], LDA can be extended to additional observed random variables for the document labels. This supervised version of LDA can be briefly summarized by the following generative process:

1. For each topic t :
 - a) Draw $\phi_t \sim \text{Dir}(\beta)$
2. For each document d :
 - a) Draw $\theta_d \sim \text{Dir}(\alpha)$
 - b) For each word w_n in document d :
 - i. Draw $t_n \sim \text{Mult}(\theta_d)$
 - ii. Draw $w_n \sim \text{Mult}(\phi_{t_n})$
 - c) Draw $y \sim \text{GLM}(t, \mu, \sigma)$

2. Latent Variable Models

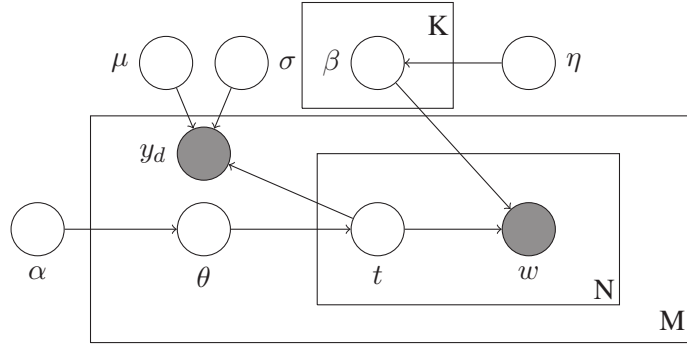


Figure 2.14.: Supervised LDA represented as graphical model in the Plate notation. In addition to standard LDA, an observed random variable y_d that depends on a given topic t is modeled as Generalized Linear Model with mean μ and variance σ .

The graphical model of supervised LDA is given in Figure 2.14. The difference to standard LDA is the observed label y_d for each document, (cf. Figure 6.2). The labels are assumed to be drawn from a Generalized Linear Model (GLM) with mean μ and variance σ . The evidence lower bound L^s is the same as for standard LDA L (see Equation 2.12) up to one term. The difference lies in the term

$$\mathbb{E} [\log p(y|t, \eta, \sigma)] = \log h(y, \delta) + \frac{1}{\delta} [\mu'(\mathbb{E}[\hat{t}y]) - \mathbb{E}[A(\mu'\hat{t})]]$$

from the GLM that is added to the bound. The terms h and A depend on the concrete GLM used and $E[\hat{t}] = \hat{\phi} = \frac{1}{N} \sum_{n=1}^N \phi_n$. This results in a new lower bound:

$$\begin{aligned} L^s(\gamma, \lambda, \phi; \alpha, \mu) &= \mathbb{E}_q[\log p(\theta, \beta, \mathbf{w}, \mathbf{t}, y|\alpha, \mu)] - \mathbb{E}_q[\log q(\theta, \beta, \mathbf{t}|\gamma, \lambda, \phi)] \\ &= \mathbb{E}_q[\log p(\theta, \beta, \mathbf{w}, \mathbf{t}|\alpha, \mu)] + \mathbb{E}_q[\log p(y|z, \mu, \sigma)] \\ &\quad - \mathbb{E}_q[\log q(\theta, \beta, \mathbf{t}|\gamma, \lambda, \phi)] \\ &= L + \mathbb{E}_q[\log p(y|z, \mu, \sigma)]. \end{aligned} \tag{2.23}$$

This motivates the interpretation of supervised LDA as regularization of standard LDA with the expectation of the label distributions under the variational distributions. This means, we optimize for variational parameters that also maximize the likelihood of the labels under the GLM. The derivatives of the variational parameters is the same as in standard LDA for γ but different for ϕ . Now in the M-step we get a new update for ϕ using the gradient:

$$\frac{\partial L}{\partial \phi_n} = E[\log \theta] + E[\log p(w_n|\phi)] - \log \phi_n + 1 + \frac{y}{N\sigma} \mu - \frac{1}{\sigma} \frac{\partial E[A(\mu'\hat{t})]}{\partial \phi_n}.$$

If we further assume that y_d are Gaussian random variables, we can derive the following updates for the GLM parameters:

$$\begin{aligned}\mu &= (E[X'X])^{-1}E[X]'y \\ \sigma &= \frac{1}{D}\{y'y - y'E[X](E[X'X])^{-1}E[X]'y\}\end{aligned}$$

with $E[X'X] = \sum_d E[\hat{t}_d \hat{t}_d']$ and $E[X] = E[\hat{t}]'$.

A different approach for supervised LDA is proposed by Zhu et al in [ZAX09]. The authors use a maximum margin approach to integrate nominal and numeric labels into topic models. Instead of assuming that the label comes from a GLM, the labels are the expected outcome of a linear classifier that belongs to a class that induce a maximum margin for the labels. This combines the generative power of LDA and the discriminative power of Support Vector Machines [Vap95].

Additional extensions of LDA to model document labels have been proposed by Lacoste-Julien et al. in [LJSJ09] and Ramage et al in [RHN09] to include categories of documents into topic models. Lacoste-Julien et al. use a transformation of the T -dimensional document-topic distribution into an l -dimensional document-category distribution that models the distribution of a document over all possible document categories. Ramage et al. on the other hand, model the assignment of a document to a category as additional Bernoulli distributed random variable.

An especially interesting type of supervision for document are time stamps. Time stamps telling when the documents were written can be used to investigate the temporal distribution of the topics. In [WM06] Wang and McCallum extend the LDA topic model such that observed time stamps (or simple time values) are assumed to be generated by the latent topics, independent of the words given the topic. This method enables a temporal alignment of the topics that reflects when the documents have been written.

Different Priors for LDA Parameters

There are usually two reasons to include additional priors on the parameters for LDA. First, due to lack of information in the data, we expect that the pure likelihood can not be estimated accurately enough and prior belief in the distribution can compensate this. Second, additional information about the data is available and we want that our topic model reflects the information. For instance information about authors of texts can be used to estimate topics such that texts from the same authors have an affinity towards certain topics.

In the literature (see for instance [WMM09]) integrating priors on the meta parameters for LDA is motivated by fully Bayesian modeling. This means, the meta parameters of LDA are also random variables that follow a certain (prior) distribution. Hence, instead of specifying α and η as fixed parameters, they are modeled for instance as Gamma distributed random variables. Alternatively, the meta parameters are modeled as $\alpha = \alpha' a$ and $\eta = \eta' b$ for fix concentration parameters α', η' and base measures a, b which again are modeled with certain prior distributions. Further approaches model the meta parameters as logistic function: $\alpha = e^a$ and $\eta = e^b$, respectively $\alpha = e^{\mathbf{a}'\mathbf{x}_d}$ and $\eta = e^{\mathbf{b}'\mathbf{x}_w}$ for k_1 additional document features $\mathbf{x}_d = [x_{d,1}, \dots, x_{d,k_1}]$ and k_2 word features $\mathbf{x}_w = [x_{w,1}, \dots, x_{w,k_2}]$ with appropriate additional prior distributions for \mathbf{a} and \mathbf{b} .

2. Latent Variable Models

Note, when modeling the meta parameters as non-random variables, we can either choose them by hand or try to find the optimal parameters by maximizing the log-likelihood for LDA with respect to the parameters. As proposed by Blei et al. [BNJ03] the latter can be easily done by Newton-like optimization. This means, the parameters are estimated directly from the data without any prior belief or information.

With the different Bayesian models of the meta parameters, different approaches are possible to efficiently integrate the priors. If we choose the prior distribution to be a conjugate prior we can easily integrate the random meta parameter out. For instance, if we model the meta parameter α as also Dirichlet distributed then the document-topic distribution θ has a hierarchical Dirichlet prior and we can integrate it out. The same is also true if we model $\alpha = \alpha' a$ and a has a Dirichlet prior and is to be integrated out.

Instead of integrating out, we can also optimize the log-likelihood with respect to the parameters \mathbf{a} or \mathbf{b} depending on the prior. This is especially interesting if we cannot integrate the corresponding parameter out. In case we define $\alpha = e^{\mathbf{a}'\mathbf{x}_d}$, respectively $\eta = e^{\mathbf{b}'\mathbf{x}_w}$, as link function for document features \mathbf{x}_d , respectively word features \mathbf{x}_w , with additional prior distributions $p_1(\mathbf{a})$ and $p_2(\mathbf{b})$, gradients of the log-likelihood of the LDA topic models can be estimated and Newton-like or general gradient based optimization methods can be used to find the optimal \mathbf{a} and \mathbf{b} .

Examples for different link functions for the meta parameters α and η with different priors are in the works of Mimno and McCallum in [MM12] and Petterson et al. in [PSC⁺10]. Mimno and McCallum propose to set $\alpha = e^{\mathbf{a}'\mathbf{x}_d}$ for document features \mathbf{x}_d with a Gaussian prior $N(0, \sigma^2)$ on \mathbf{a} with mean 0 and variance σ . For document features like author indicator features (binary vectors with 1 at a certain component indicating the author of the document), the optimal \mathbf{a} is estimated by maximizing the log-likelihood via gradient ascent with the partial derivatives for the features f and topics t

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{a}_{t,f}} &= \sum_d x_{df} e^{\mathbf{a}'_t \mathbf{x}_d} (\Psi(\sum_t e^{\mathbf{a}'_t \mathbf{x}_d}) - \Psi(\sum_t e^{\mathbf{a}'_t \mathbf{x}_d} + n_d)) \\ &\quad + \Psi(e^{\mathbf{a}'_t \mathbf{x}_d} + n_{d,t}) - \Psi(e^{\mathbf{a}'_t \mathbf{x}_d}) - \frac{\mathbf{a}_{t,f}}{\sigma^2}, \end{aligned}$$

with Ψ the first derivative of the logarithm of the Gamma function, named the Digamma function. For LDA with Gibbs sampler, the authors propose to perform this gradient ascent every other iteration via the standard non-linear optimization solver BFGS [LN89].

Petterson et al. on the other hand define a link function on the meta parameter for the topic-word distribution prior, hence $\eta = e^{\mathbf{b}'\mathbf{x}_w}$. By this, we can include additional information about the words via word features \mathbf{x}_w . Further, the authors propose a special prior on \mathbf{b} that includes similarity information $\text{sim}(w, w')$ about the words by a graphical model. The optimal parameters \mathbf{b} are estimate similar to the approach by Mimno and McCallum via gradient based optimization. The major difference is the special prior

$$p(\mathbf{b}) = e^{\frac{-1}{2\sigma^2} \sum_{w,w',t} \text{sim}(w,w')(b_{t,w} - b_{t,w'})^2}.$$

In [WXK10], Wahabzada et al. propose to integrate relations among documents via Gaussian Processes. The link function is defined as $\alpha = e^{f(d)}$ with a Gaussian Process prior on $f(d)$. Yuan et al. use in [YZX12] the approach by Mimno and McCallum to define a topic model over geographical regions with feature information about the regions and certain points-of-interest. In [He12], He proposes to define the meta parameter for the word-topic distribution as linear combination of sentiment specific word prior information. These approaches show that priors can be used to integrate arbitrary information about documents and words into topic models.

2.3.3. Further Topic Models

In the previous sections, we described a fixed model. Latent Dirichlet Allocation uses multinomial distributions with Dirichlet priors as depicted in Figure 6.2. Setting the meta parameters as link function or adding additional priors on the parameters did not change this model. On the other hand, there are many different approaches to slightly change LDA to model the documents and words differently.

We can, for instance, change the assumption in LDA that the document-topic and the topic-word distributions are multinomial distributions with Dirichlet priors. Instead of the Dirichlet priors different prior distributions can be used. This is different to the previous section where we used priors on the meta parameters of the Dirichlet distributions. In [NBB11a] for instance, Newman et al. propose structural priors based on side information instead of the Dirichlet priors. Given covariance information about the words in a matrix $K \in \mathbb{R}^{V,V}$ they propose the following prior on the topic-word distribution:

$$p(\beta|K) \propto (\beta'K\beta)^\nu.$$

The prior allows the inclusion of correlation information about words into topic models. The integration of the prior into the topic modeling is done by maximizing the posterior of this new model, resulting in the following word-topic distribution:

$$p(\mathbf{d}|t) \propto \prod_n \beta_{t,w_n}^{n_{t,w_n}} (\beta_t'K\beta_t)^\nu.$$

Another prominent adaptation of standard LDA is the so called Correlated Topic Model. In order to also model correlation between topics (not between documents or words), Blei and Lafferty propose in [BL06a] to model the topics as parametrized multinomial distribution:

$$p(t|\mathbf{a}) \propto e^{\mathbf{a}'\mathbf{x}}.$$

The mapping into the probability simplex for the multinomial distribution θ is then via

$$\theta_i = \frac{e^{a_i}}{\sum_j e^{a_j}}.$$

2. Latent Variable Models

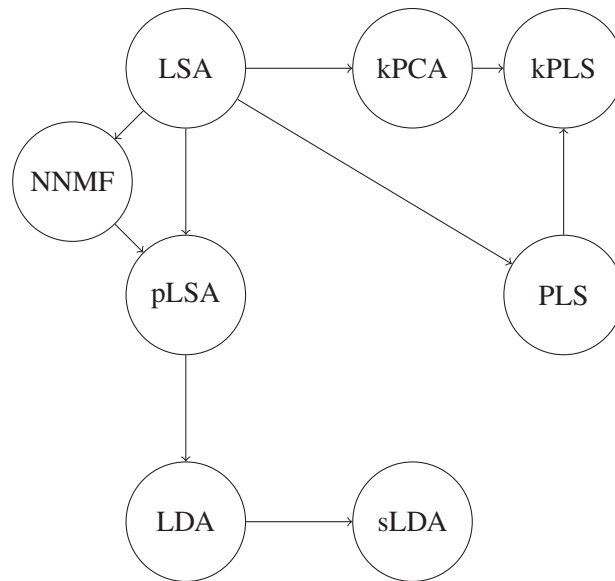


Figure 2.15.: Schematic view of the relations of the most prominent factor and topic models. In contrast to LSA, NNMF puts non-negativity constraints on the factors. A probabilistic interpretation of NNMF leads to pLSA. In pLSA we perform a decomposition of the joint probability of words and concepts while in LSA the decomposition is on the term-document matrix. Via kernels, kPCA can extract non-linear factors in contrast to the linear factors from LSA. The methods PLS and kPLS extend the extraction of latent factors (linear in LSA, non-linear in kPCA) to consider given document labels. Integrating Dirichlet priors on the probability distributions of pLSA results in LDA. Jointly modeling words, concepts and document labels extends LDA to sLDA.

The additional normally distributed prior on the parameters \mathbf{a} allows the estimation of a mean and a covariance between topics. The estimation of the topics, the topic mean and the topic covariance can be done by variational inference. The only caveat is that the proposed normal prior is not conjugate to the multinomial distribution θ . To solve this, Blei and Lafferty propose to bound the expectation $\mathbb{E}_q[p(t|\mathbf{a})]$ via Taylor approximation.

These are only two examples of the several approaches to alternate LDA to enhance the expressiveness of the topic model. Further works are for instance Dynamic Topic Models [BL06b] that also model the evolution or temporal order of topics. In [LM06] hierarchies of topics are included; in [SN10] the topic-word distribution also consider Zipf's law; in [DE09] word probabilities also account for burstiness and in [BGBZ07] the topic-word distribution is replaced by a topic-path distribution of semantic nets like WordNet to name only a few additional approaches.

To conclude, in Figure 2.15 we illustrate the relations between the latent variables models in a schematic way. The connections between the different latent variable methods is illustrated as graph.



3. Evaluation Methods

We distinguish between qualitative and quantitative methods to evaluate the quality of the latent variable models for a given linguistic task as introduced in Chapter 1. Such methods shall indicate the value of the extracted factors and topics for linguistic tasks. While qualitative evaluation methods show how useful the extracted information about latent topics or factors are for linguistic research, the quantitative evaluation methods provide mechanisms to automatically compare the models.

3.1. Qualitative Evaluation Methods

In practice, the factor and topic models are used qualitatively. Experts interpret results of the models by exploring the factors and topics. For the linguistics tasks for example, the results of latent variable modeling are mostly manually investigated. For example, if we are interested in usage patterns of expression and words in context and over time, we need methods to evaluate

3. Evaluation Methods

	LDA	LSA	PLS	NNMF	kPLS/kPCA
$p_t(w)$	Mult(β_t)	$N(R_i, 1)$	$N(v^t, 1)$	$N(V_t)$	$N(\phi_t, 1)$
$p_d(t)$	Mult(θ_d)	Mult($L_d, 1$)	Mult($l_d, 1$)	Mult(W_d)	$N(\psi, 1)$
$p(\tau_d)$	Beta(τ)/SG(τ)	-	$N(\tau, 1)$	-	$N(\tau, 1)$

Table 3.1.: Probability distributions of words, documents and time stamps for visualization. For factor models, we use the normal distribution N based on distances as surrogate measure for the word probability distribution $p(w|t)$ and the document distribution $p(t|d)$ with respect to a given latent variable t . For topic models, these probability distributions are explicitly modeled as multinomial distributions Mult during model inference.

the results of a latent variable model in terms of the words and documents. Rather, than abstract numbers that describe the results, we are interested in how explanatory the factors, respectively the topics, are. A good format and a visualization of the results is needed to help evaluating the models by linguists. There are several possible ways to visualize the results of factor and topic models. In the literature there are usually the following aspects considered: First, how can we show the tendency of words and documents to certain topics. Second, how can show the distribution of the topics over the words and documents. Finally, how can we show the distribution of topics, words and documents over time. The latter is important for diachronic linguistics.

3.1.1. Ranking Lists and Word Clouds

One straightforward way to qualitatively evaluate the factors, respectively the topics, is to inspect the "importance" of the words given a latent variable. This importance shall measure how much influence words have for given factors or topics. Using such a measure, we can rank all words such that the most important words have highest rank. As concrete importance measure, we use the probability of a word w for a given latent variable t noted as $p(w|t)$.

For topic models, we can quite easily measure the importance of words given a topic by the probabilities estimated during inference. This can directly be read off from the parameters $\beta_{w,t}$ for the multinomial distribution of the words for topic t , (see 2.3.2). Hence,

$$p(w|t) = \beta_{w,t}.$$

For a factor model this is not straightforward. In order to estimate the importance of a word for a given factor, we need a surrogate measure that can be used as probabilities: Let \mathbf{v}^i be the i_{th} factor. This can be the i_{th} column from R in LSA or V in NNMF, the i_{th} loadings vector from PLS or the projection onto i_{th} component in a Hilbert space by kernel PCA or kernel PLS. A word represented as Word-Vector can be associated with a similarity value based on the distance in the Euclidean (or Hilbert) space to \mathbf{v}^i . This can be seen as the reconstruction error for the word in the vector space after projection.

Formally, the distance of a Word-Vector \mathbf{w}^j for word w_j to a latent factor is defined as

$$\|P_i \mathbf{w}^j\|_2^2,$$

for

$$\mathbf{w}^j = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

the standard unit vectors that corresponds to word w_j and the projection matrix P_i onto the factor \mathbf{v}^i :

$$\mathbf{v}^i = \begin{bmatrix} v_1^i \\ \vdots \\ v_V^i \end{bmatrix}.$$

This is the length of the orthogonal projection of \mathbf{w}^j onto \mathbf{v}^i and is equal to the component in \mathbf{v}^i that corresponds to the dimension spanned by the word w_j in the vector space. We note this value as \mathbf{v}_j^i . Hence, \mathbf{v}_j^i is the value in vector \mathbf{v}^i at position j that corresponds to the dimension in the VSM of word w_j .

Since we are only interested in the order of words based on their importance, we can apply any order preserving transformation. This motivates the definition of word probabilities $p(w_j|t = i)$ based on this distance measure. For factor models, we define the probability of a word w_j for a given factor \mathbf{v}^i as proportional to the distance of the word to the factor in the vector space:

$$p(w_j|t = i) \propto \|P_i \mathbf{w}^j\|_2^2.$$

A simple word probability based on distances in the vector space is the normal distribution with mean 0:

$$p(w_j|t = i) = \frac{1}{2\sigma} e^{-\frac{1}{2\sigma^2} \|P_i \mathbf{w}^j\|_2^2}.$$

Now, we can define ranking lists of word associated with each factor or topic as

$$V_t = (w_1, \dots, w_k)$$

such that

$$\forall m \leq n < k : p(w_m|t) \geq p(w_n|t).$$

This ranking list contains the top- k words in decreasing order of the word probability $p(w_i)$. With similar consideration, we can also define rankings of documents for each topic. In LDA for instance the document-topic distribution $p(d|t)$ with parameter θ can be directly used as importance of a document for a topic. In factor models, we use a similar surrogate measure as for the words based on the distance of the document represented as Word-Vector to the factor:

$$p(t = i|d) = \frac{1}{2\sigma} e^{-\frac{1}{2\sigma^2} \|P_i \mathbf{w}_d\|_2^2}.$$

A particular visualization of the word with respect to its importance is now easily done. Via world clouds for example, we can visualize the importance of words based on $p(w|t)$. Each

3. Evaluation Methods

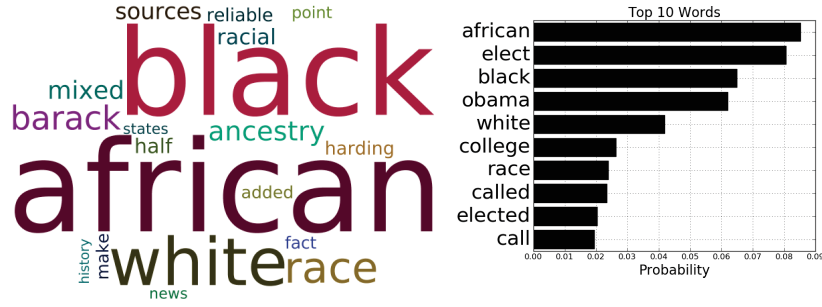


Figure 3.1.: Visualization of the most important words for a given factor or topic. Left: a Word Cloud from the highest ranked words from a topic model on Wikipedia talk pages containing the word “president”; The larger the size of the word, the more frequent is this word. right: a sorted list of the highest ranked words (y-axis) from a topic model about presidents with frequency values on the x-axis.

of the top- k words from the ranking list is written in a figure with size proportional to $p(w|t)$. On the left in Figure 3.1, we see a Word Cloud from a topic model containing topics about presidents. Besides Word Clouds, we can also just list the top- k words in decreasing order of importances in the concrete values of $p(w|t)$ can be additionally plotted as histogram. This can be seen on the right in Figure 3.1.

3.1.2. Temporal Distribution

Ranking lists and Word Clouds can visualize the word distributions for factors or topics. For the temporal distributions of the documents with respect to the factors or topics, we need to display the course of the importance of the latent variables over time. The amount of a certain factor or topic in a given time can be estimated by grouping documents by time and averaging the document-topic proportions. Analogue to the importance of a document to a certain latent variable, the document-topic proportion tells how much present a certain topic is in a document.

Each document d has its time stamp τ_d . Grouping these values into e intervals

$$[0, \tau_1], [\tau_1, \tau_2], \dots, [\tau_{e-1}, \tau_e],$$

we assign the documents to the corresponding intervals, hence $d \rightarrow [\tau_i, \tau_{i+1}]$ with $\tau_i \leq \tau_d \leq \tau_{i+1}$. Now, we can average the $p(t|d)$ in each interval to get a histogram of topic proportions over time.

Similar to the word importance measures from the last section the document-proportion can be easily derived. For topic models the document-topic proportions are the multinomial distributions $\text{Mult}(\theta_d)$. Hence

$$p(t|d) = \theta_{d,t}.$$

For the factor models we need again the surrogate measures that can be used to estimate probabilities of documents for certain topics based on distances in the vector space of the Bag-of-Words. This can be done as discussed in previous sections. An other approach is to apply a

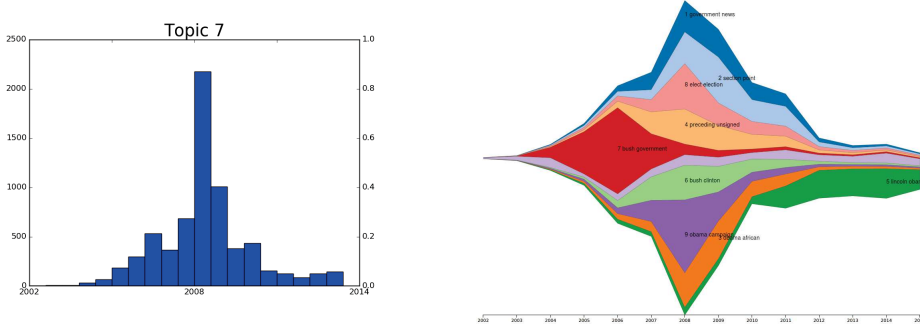


Figure 3.2.: Visualization of the distribution of topics over time. Left: Histogram of topic proportions for a given topic over time from a topic model on Wikipedia talk pages about presidents. On the x-axis, the years from 2002 to 2014 are marked. On the y-axis, the amount of topic 7 by the number of document assigned to this topic for each of these years in the Wikipedia talk pages is plotted. Right: Plot of stacked time series derived from histograms of topic proportions for all topics over time (x-axis).

multinomial transformation to map the factors into a multinomial distribution. For factor models the document-factor proportions are T -dimensional vectors ω_d for

$$\mathbf{w}_d \approx \sum \omega_{di} \mathbf{v}^i.$$

Depending on the used factor model, the document-factor proportion can be directly derived from the factorization. For LSA, the document-factor proportion is the left singular vector (multiplied by its singular value) l_d . For NNMF, the document-factor proportion is the vector ω_d from the factorized term-document matrix W . These proportion vectors can be transformed into a probability, by projecting them onto the probability simplex:

$$\omega_d \rightarrow \left(\frac{\omega_{d1}}{\sum_j \omega_{dj}}, \dots, \frac{\omega_{dT}}{\sum_j \omega_{dj}} \right).$$

This is a multinomial distribution just like the document-topic distribution θ_d from LDA. Now, we define the probability of a factor t for a given document d as

$$p(t|d) = \frac{\omega_{dt}}{\sum_j \omega_{dj}}.$$

Again, these proportion values can be aggregated, grouped by documents that fall into a certain time interval. Since the proportions are multinomial distribution for factor and topic models, we can compare them and use the same visualization.

In Figure 3.3, we show how the histograms over time can be visualized to qualitatively evaluate topic distributions over time. On the left, we show that a simple plot of the histogram can be used to inspect a single topic or factor for its distribution over time. On the right, we show the

3. Evaluation Methods

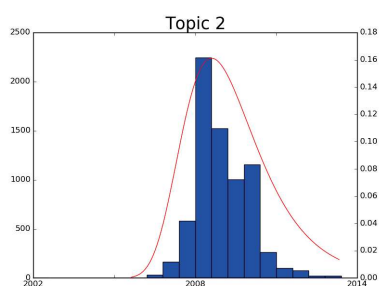


Figure 3.3.: Visualization of the distribution of a topic over time as histogram (on the left y-axis) of the number of document assigned to topic 2. From the year 2002 to 2014 (x-axis), the number of texts with topic 2 is counted (left y-axis). Additionally on the right y-axis, we plot the density of the time stamps fitting the corresponding time stamps for this topic.

yearly sub-view of from the DRF-Browser by Andrew Goldstone¹. This visualization enables to compare distributions of several factors or topics over time at once. Here, the histograms are used as time series. These series are plotted and stacked on top of each other.

Since we can also explicitly model time stamps with a certain distribution by factor and topic models, we can further use this estimated distribution for qualitative evaluation. As discussed in the last chapter, for topic models we can model the time stamps as Beta distributed. Now, the density of the estimated temporal distribution for certain topics can be visually inspected to investigate the course of the time stamps and the likely continuation of the topic in the future. In Figure 3.3, we show the density of an estimated distribution of time stamps together with the corresponding histogram of time stamps for a topic from a topic model about presidents.

3.1.3. Geometric Interpretation

A geometric interpretation of the latent variable models, especially the factor models, allows for a visualization of the underlying document representation. For factor models we can visualize the Word-Vector and the words in a vector space. Since in the VSM, we associate the standard unit vectors that span a vector space with the words from a vocabulary of a corpus, we can inspect individual words by visualizing the Word-Vector in the corresponding dimensions. Each word is identified by a corresponding unit vector w^i . The position of the vectors in the vector space shows relations between words. By this, we can explore possible correlations between the words. Totally independent words, for examples, will be orthogonal in the corresponding dimensions. Co-occurring words, on the other hand, will have a clear functional relation in the vector space. In Figure 3.4 on the left, we visualize the two dimensional subspace spanned by the words *good* and *professional* in the VSM for a corpus containing reviews about books. The words are highly correlated as seen by the linear relation between the words in the vector space. Besides the location of the Word-Vector and the words in the vector space in the VSM, the distance between the words helps to interpret the factors. The space spanned by the factors in

¹<https://agoldst.github.io/dfr-browser/>

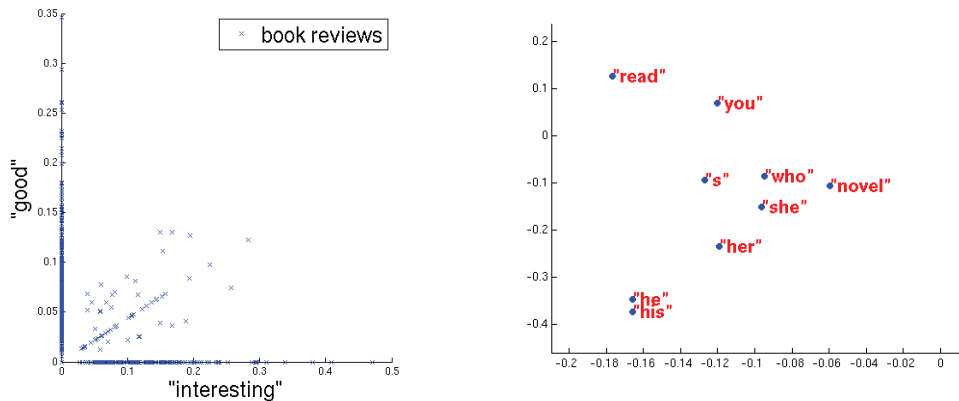


Figure 3.4.: Left: Subspace spanned by the words "good" and "interesting" in the vector space spanned by the Word-Vectors with frequency value. The axis tell the frequencies of the words in the book reviews. Right: Words projected onto the first two components extracted from a term-document matrix via an SVD.

the vector space shall make semantically related words more similar in terms of lower distance. In Figure 3.4 on the right, we plot the words projected onto the first two components from an SVD of the Term-Document Matrix from reviews about books. We see, that pronouns are very close in this subspace. This visualizes the semantically related words as group of words with low distance in the vector space.

3.2. Quantitative Evaluation Methods

Quantitative evaluation methods measure the quality of the proposed models on a numeric scale. In contrast to the qualitative evaluation methods, we can automatically compare the models on large data sets. We can perform large numbers of evaluations on many different data sets to assess the quality of the models.

3.2.1. Coherence Measures

Frequently used quantitative evaluation methods are based on the relations of the highest ranked words in each topic. The coherence measures estimate how well the model fits an as coherent expected outcome. The definition of this expected coherent outcome is usually based on user studies and experience with topic modeling in practice. A fundamental assumption for topics or factors to be coherent is based on the top ranked words. Each topic is associated with a value how present this topic is for given words. In LDA, this value is from the multinomial distribution β_t , in LSA, it is the length of the projection of this word onto the latent factor in the vector space spanned by the Word-Vectors. Ranking the words for each topic results in a compact representation of the each topic.

For T latent variables with the corresponding top k words in ranking lists $V_t =$

3. Evaluation Methods

$\{w_{1t}, \dots, w_{kt}\}$ with respect to each latent variable that is extracted by a latent variable model, the overall coherence measure is the mean over individual coherence values $U(V_t)$:

$$U(V) = \frac{1}{T} \sum_{t=1}^T U(V_t).$$

To estimate the individual values for a given latent variable model, we use several coherence measures that have been proposed in the literature. All measures use statistics of co-occurring words from an additionally given reference document collections like Wikipedia articles. In all later experiments, we use the tool Palmetto² to estimate the individual coherence values. For a detailed description of the tool and the quality measures see [RBH15]. In the next subsections, we describe the coherence measures mostly used in literature for topic models. Nonetheless, these methods are also applicable to factor models.

UMass

In [MWT⁺11], Mimno et al. propose a topic coherence measure that depends on co-occurrences of words. Based on user studies, they show that this measure corresponds well with the top ranked topics by the users. In the literature the measure is called the U_{Mass} measure and is defined as

$$U_{Mass}(V_t) = \sum_{m=2}^k \sum_{l=1}^m \log \frac{D(w_{mt}, w_{lt}) + 1}{D(w_{lt})}. \quad (3.1)$$

The measure is the sum of the log-ratios of the by 1-smoothed co-occurrence frequency of any two ordered words in the top ranked list, $D(w_{mt}, w_{lt})$, and the document frequency of the lower ranked word, $D(w_{lt})$.

Pointwise Mutual Information

The authors in [NLGB10] introduce Pointwise Mutual Information (PMI) as measure for topic coherence. The PMI is the log-ratio of the joint probability of two random variables and the product of their marginal probabilities. It measures how likely two random variable are jointly distributed and not independently distributed. The PMI of two words w_1 and w_2 is defined as

$$PMI(w_1, w_2) = \log \frac{p(w_1, w_2)}{p(w_1)p(w_2)}.$$

The PMI can be interpreted as how much likely the two words w_1 and w_2 appear together in contrast to how likely they appear alone.

For a latent topic, respectively factor t and the top k ranked words V_t , the PMI is defined as

$$PMI(V_t) = \frac{1}{(k-1)k/2} \sum_{m < n}^k \log \frac{p(w_{mt}, w_{nt})}{p(w_{mt})p(w_{nt})}. \quad (3.2)$$

²<https://github.com/AKSW/Palmetto>

Aletras and Stevenson propose in [AS13] to also use Normalized Pointwise Mutual Information (NPMI) to estimate the coherence of the topics. The NPMI is the PMI divided by the negative log probability of the two words appearing together. The reason to use NPMI is twofold. First, NPMI is normalized between -1 and 1 . Second, low frequencies of the words are less critical. Especially the second reason is important, since small outliers can result in very small joint probabilities that overtake the whole coherence measure. Formally the NPMI is defined as

$$NPMI(V_i) = \frac{1}{(k-1)k/2} \sum_{m < n} \frac{\log \frac{p(w_{mt}, w_{nt})}{p(w_{mt})p(w_{nt})}}{-\log p(w_{mt}, w_{nt})}. \quad (3.3)$$

Temporal Coherence

Similar to the coherence of the top ranked words, we estimate the temporal coherence as distance of the distribution of the time stamps associated with a latent variable, with the distribution of the time stamps for the top words over all documents in the corpus. This is a new quality measure for factor models with temporal information. We assume that the documents containing the top words from latent variables approximate the content of the underlying concept. The temporal difference of the time stamps of these documents indicates how well this latent information captures the true temporal dynamics in the corpus. Here, we concentrate only on latent topic models, but latent factor models can be treated in the same way. A topic is temporal coherent if the estimated distribution of the time stamps in this topic is similar to the temporal distribution of the time stamps for the top words in the whole corpus. The documents that contain the top two words approximate the semantic behind the topic or factor. Hence, documents containing the top two words in the corpus can be used as coherence reference. In all experiments (cf. Chapter 5), we perform all our empirical estimations based on the top two words in the topics, but higher order n-grams are also possible.

The empirical distributions of the time stamps of the topics and the top words in the corpus are estimated by histograms h_t and h_{w_1, w_2} . For a topic t , the empirical probability of the time between two time stamps τ_1 and τ_2 can be approximated by

$$P([\tau_1, \tau_2] | t) = p(\tau_2 | t) - p(\tau_1 | t) \propto \sum_{\tau} I_{\tau_1 \leq \tau < \tau_2}(\tau) n_{\tau, t}.$$

For $n_{\tau, t}$ the number of tokens assigned to topic t from a document with time stamp τ and the indicator function

$$I_{\tau_1 \leq \tau < \tau_2}(\tau) = \begin{cases} 1, & \tau \in [\tau_1, \tau_2] \\ 0, & \text{else.} \end{cases}$$

Now, we define the histogram of the temporal distribution of topic t as function $h_t : \mathbb{N} \rightarrow \mathbb{N}$ such that

$$h_t(\tau_1, \tau_2) = \sum_{\tau} I_{\tau_1 \leq \tau < \tau_2}(\tau) n_{\tau, t},$$

for a given number of intervals $[\tau_1, \tau_2], \dots, [\tau_{e-1}, \tau_e]$.

For two words w_1 and w_2 for topic t , the empirical probability can be approximated by

$$P([\tau_1, \tau_2] | w_1, w_2) = p(\tau_2 | w_1, w_2) - p(\tau_1 | w_1, w_2) \propto \sum_{\tau} I_{\tau_1 \leq \tau < \tau_2}(\tau) n_{w_1, w_2, \tau}$$

3. Evaluation Methods

for $n_{w_1, w_2, \tau}$ the number of tokens in the documents that contain both words w_1 and w_2 in the corpus with time stamp τ . The histogram of the temporal distribution of the words w_1 and w_2 in the corpus is the function $h_{w_1, w_2} : \mathbb{N} \rightarrow \mathbb{N}$ such that

$$h_{w_1, w_2}(\tau_1, \tau_2) = \sum_{\tau} I_{\tau_1 \leq \tau < \tau_2}(\tau) n_{w_1, w_2, \tau}.$$

There are several distance measures possible. We propose to use the Minkowski distance to estimate how much the distributions over the time stamps differ based on histograms. The Minkowski distance of two histograms for topic t and the corresponding top two words w_{1t}, w_{2t} is defined as

$$D(h_t, h_{w_{1t}, w_{2t}}, p) = \sqrt[p]{\sum_i |h_t(\tau_i, \tau_{i+1}) - h_{w_{1t}, w_{2t}}(\tau_i, \tau_{i+1})|^p}.$$

Using $p = 2$ is the Euclidean distances and $p = 1$ is the l_1 distance. In all our experiments, we use $p = 1$.

3.2.2. Likelihood

The coherence measures estimate the quality of latent variable models based on statistics from different document collections and user information. To estimate how good a factor or topic model fits the corpus we estimate the likelihood of the data under this model. Depending on the specific model, we can directly estimate the likelihood or we need special assumptions. For topic models, the likelihood of a set of test documents in corpus C_{te} given a topic model by its parameters is

$$p(C_{te} | \alpha, \beta) = \prod_{d \in C_{te}} p(\mathbf{d} | \alpha, \beta).$$

This involves difficult normalization constants as described above. In the next subsection, we describe efficient sampling methods that estimate the probabilities.

For factor models, the likelihood is more difficult to estimate. We cannot directly get probabilities of the words from the model. Instead, we must use a language model that depends on the factor model. This language model defines the probabilities of the words, given the factors \mathbf{v}^i . Formally, this is

$$p(C_{te} | \mathbf{v}^i) = \prod_{\mathbf{w}_d \in C_{te}} p(\mathbf{w}_d | \mathbf{v}^i).$$

At the end of this section, we give an example to estimate $p(\mathbf{w}_d | \mathbf{v}^i)$ for a given factor model.

Importance Sampling

A simple and straight forward way to estimate $p(w)$ for topic models is the use of Importance Sampling. Importance Sampling uses re-weighted samples from a simpler distribution such

that the weighted samples approximately follow the distribution $p(w)$. Consider the following reformulation of the probability:

$$p(w) = \sum_t p(w, t) = \sum_t \frac{p(w, t)q(t)}{q(t)} = \mathbb{E}_q \left[\frac{p(w, t)}{q(t)} \right] \sim \frac{1}{S} \sum_s \frac{p(w, t^s)}{q(t^s)}.$$

This means, we approximate the word probability $p(w)$ by S weighted samples t^s from a so called proposal distribution $q(t)$. This approximation depends on $q(t)$ and its deviation from $p(t)$. For the concrete proposal distribution, it should be simple to sample from and its support must be a superset of the support of $p(t)$. Wallach et al. [WMSM09] propose for instance to use $p(t|\alpha) \sim \text{Dir}(\alpha)$ as proposal distribution.

Harmonic Mean Method

Another approach to estimate the probability $p(w)$ by sampling methods is to approximate it by harmonic means of conditionals $p(w|t^s)$. Consider the following reformulations using the Bayes Rule:

$$p(w)p(t|w) = p(w|t)p(t).$$

Dividing this equation by $p(w|t)$ and assuming $p(w|t) \neq 0$, we get

$$p(w) \frac{p(t|w)}{p(w|t)} = p(t).$$

Since this equation is true for every t , we can sum over all latent variable on the left and the right hand side to get

$$p(w) \sum_t \frac{p(t|w)}{p(w|t)} = \sum_t p(t).$$

Since $p(t)$ is a probability, we have: $\sum_t p(t) = 1$. Now, we can reformulate the last equation such that

$$\frac{1}{p(w)} = \sum_t \frac{p(t|w)}{p(w|t)} = \mathbb{E}_{p(t|w)} \left[\frac{1}{p(w|t)} \right].$$

Consequently, the reciprocal value of the word probability can be expressed as the expected value of $\frac{1}{p(w|t)}$ from the probability $p(t|w)$. Finally we can approximate the expectation by samples $t^s \sim p(t|w)$:

$$\mathbb{E}_{p(t|w)} \left[\frac{1}{p(w|t)} \right] \sim \sum_s \frac{1}{p(w|t^s)}.$$

This results in the following estimator:

$$p(w) \sim \frac{1}{S} \sum_s (p(w|t^s)^{-1})^{-1}.$$

The right hand side of the last equation is the Harmonic Mean of $\{p(w|t^s)\}_{s=1}^S$, that is where the name Harmonic Mean Methods comes from. Again, as proposal distribution $p(t|w)$ we can use $\text{Dir}(\alpha)$.

3. Evaluation Methods

Sequential Monte Carlo

As proposed by Wallach in [WMSM09], Sequential Monte Carlo methods can be used to estimate the likelihood of a topic model. For a sequence of words from a hold-out test data set, the probability of the test words w is

$$p(\mathbf{d}|\alpha, \beta) = \prod_m p(w_m|\mathbf{d}_{<m}, \alpha, \beta).$$

A Sequential Monte Carlo algorithm to estimate the likelihood of a held-out data set for a given topic model can be defined in the following way: Given a new document d as sequence of tokens $\mathbf{d} = (w_1, \dots, w_N)$, we re-sample topic proportions for each token w_m in \mathbf{d} , given all tokens before, $\mathbf{d}_{<m} = (w_1, \dots, w_{m-1})$, using the point estimate of the topic-word distributions. To compensate the uncertainty in these estimates for a single document, we keep M independent samples. These samples are called particles. For the m_{th} word in the sequence, the probability is

$$p(w_m|\mathbf{d}_{<m}, \alpha, \beta) = \sum_{i=1}^T p(t_i|\theta)p(w_m|\mathbf{d}_{<m}, t_i, \beta) \quad (3.4)$$

$$= \sum_{i=1}^T p(t_i|\theta, \mathbf{d}_{<m})p(w_m|\mathbf{d}_{<m}, t_i, \beta) \quad (3.5)$$

$$= \sum_{i=1}^T p(t_i|\theta, \mathbf{d}_{<m})p(w_m|t_i, \beta) \quad (3.6)$$

$$= \sum_{i=1}^T \frac{n_{d,i,<m} + \alpha_k}{n_i + \sum_{k'} \alpha_{k'}} \beta_{t_i, w_m}. \quad (3.7)$$

This is a mixture of multinomial distribution with Dirichlet prior $\text{Dir}(\eta)$, with mixing weights $p(t_i|\theta)$ for Dirichlet ($\text{Dir}(\alpha)$) distributed $p(t|\theta)$. Due to the independence assumption in LDA, we get from Equation 3.5 to Equation 3.6. In Equation 3.7, we apply the definition of the Dirichlet distributed multinomial distribution $p(t|\theta)$ and the definition of the point estimated of the topic-word distribution $p(w|t)$ from a trained LDA topic model.

We apply Sequential Monte Carlo Methods using particle learning (PL) methods as proposed by [SB13] and by [NLS14]. To get an estimate for the topic weights, we use aggregated counts of topic assignments for topics i , n_i , respectively for the document d , $n_{d,i}$. For $m = 1, \dots, M$, we use aggregated counts $n_{d,i,<m}$, with count assignments for all tokens up to the m_{th} , sampled iteratively from

$$p(t = i|w_m, t_{<m}) \propto \frac{\alpha_k}{\sum_{k'} \alpha_{k'}} \beta_{m,i}$$

and collected as particles. We re-sample for topic proportions for the documents, but use the point estimate for the word distribution in each topic from LDA.

Then, we sample for each particle and its corresponding aggregated counts, topic assignments and add them to these counts. This means, we have Z estimates of the aggregated counts and

consequently can estimate Z times $p(w_m)$. This models the uncertainty about the assignment by Z particles.

We define particles $T_{m,z} \sim p(t|w_1, \dots, w_m, \tau_d, \beta)$ for $z = 1, \dots, Z$. The $T_{m,z}$ are iteratively sampled such that $T_{N,z} \sim p(t|\mathbf{d}, \tau_d, \beta)$.

For supervised topic models by additional random variables that depend on the latent topics, we can easily extend to Sequential Monte Carlo method from above to estimate the likelihood of hold-out documents with addition document features like time stamps:

$$p(w_m, \tau_d | \mathbf{d}_{<m}, \alpha, \beta) = \sum_{i=1}^T p(t_i | \theta) p(w_m, \tau_d | \mathbf{d}_{<m}, t_i, \beta) \quad (3.8)$$

$$= \sum_{i=1}^T p(t_i | \theta, \mathbf{d}_{<m}) p(w_m | \mathbf{d}_{<m}, t_i, \beta) p(\tau_d | t_i) \quad (3.9)$$

$$= \sum_{i=1}^T p(t_i | \theta, \mathbf{d}_{<m}) p(w_m | t_i, \beta) p(\tau_d | t_i) \quad (3.10)$$

$$= \sum_{i=1}^T \frac{n_{d,i,<m} + \alpha_k}{n_i + \sum_{k'} \alpha_{k'}} \beta_{t_i, w_m} p(\tau_d | t_i). \quad (3.11)$$

This is a mixture of multinomial distributions with Dirichlet prior $\text{Dir}(\eta)$, with mixing weights $p(t_i | \theta) p(\tau_d | t_i)$ for Dirichlet ($\text{Dir}(\alpha)$) distributed $p(t | \theta)$ and Shifted-Gompertz distributed $p(\tau | t)$. This density is analogue to Equation 3.4 using additional time stamps. The difference lies in the integration of the temporal distributions. This is easy due to the independence assumption in temporal topic modeling. We see this as we get from Equation 3.8 to Equation 3.9.

Besides the joint likelihood of the words and the time stamps, we are also interested in the conditional likelihood. The conditional likelihood $p(\mathbf{d} | \tau_d)$ is the likelihood of the sequence of words in a test document given the time stamp. This conditional likelihood estimates the likelihood of words from the documents at the time of the document. This measure focuses on the quality of the estimated word distribution. Due to the independence assumption in the topic models, we have the following conditional probability of a sequence of words in a document given the corresponding time stamp:

$$p(\mathbf{d} | \tau_d) = \prod_n p(w_n | \tau_d).$$

The partial conditional probabilities can be calculated via

$$p(w_n | \tau_d) = \frac{p(w_n, \tau_d)}{p(\tau_d)}.$$

The joint probability $p(w_n, \tau_d)$ is estimated as in Equations 3.7 and the probability of the time stamp τ_d is

$$p(\tau_d) = \sum_t p(\tau_d | t).$$

3. Evaluation Methods

Language Model from Distances

Unlike for topic models, factor models assume that the Word-Vectors can be expressed as a linear combination of factors in a Euclidean space. Hence, we have no notion of probability distributions. To estimate the quality of a factor model, an intuitive measure is the distance of the Word-Vector in the Term-Document Matrix X to their low-dimensional feature representation induced by the factors. As discussed above for qualitative evaluation this indicates the relation of the factors to the original data. In LSA or PLS for example, the Word-Vectors are projected into the low-dimensional feature representation PX in a corresponding subspace via a projection matrix P . Then, l_2 -reconstruction error for example is

$$\|X - PX\|_2^2.$$

In NNMF on the other hand, the Word-Vectors are expressed as positive linear combination of non-negative factors such that $X \approx WV$. Then, l_2 -reconstruction error for example is

$$\|X - WV\|_2^2.$$

Last, assuming more complex representations of the documents than the Bag-of-Words approach with Word-Vectors, via high (or infinite dimensional) feature vectors in Hilbert spaces as in kernel kPCA or kPLS, the l_2 -reconstruction error for example is

$$\sum_i \|\phi(\mathbf{w}_{d_i}) - P_U(\phi(\mathbf{w}_{d_i}))\|_2^2,$$

for $\phi(\mathbf{w}_{d_i})$ the feature maps of document d_i into a RKHS and $P_U(\phi(\mathbf{w}_{d_i}))$ the projection operator onto subspace U .

Measuring the quality of the factor models by the reconstruction error in the Euclidean or Hilbert space has the disadvantage that we cannot compare these values to the quality measure extracted for the topic models.

To compare factor models with topic models, we need a common measure in the same space. In factor models, the factors are vectors spanning a subspace in the vector space spanned by the Word-Vectors. In topic models, the topics are represented by multinomial distributions drawn from a probability simplex by a Dirichlet prior distribution.

The question is now, should we explicitly model the factors as probabilities or the multinomial distributions of the topics as vectors in a Euclidean space to compare the qualities of factor and topic models. We can either interpret the Word-Vectors as likelihoods of the words, using term frequencies for instance. On the other hand, we can also model the likelihood of the words based on the embedding of the Word-Vectors in the space spanned by the factors. This defines a language model based on similarities of words in the subspace spanned by the factors.

Again, we use the sequential model proposed by [CJ98] for the definition of the document probabilities for a sequence of Word-Vectors $\{\mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_M}\}$:

$$p(\{\mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_M}\}) = \prod_{i < M} p(\mathbf{w}_{d_i} | \mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_{i-1}}).$$

The partial conditional probabilities are defined as

$$p(\mathbf{w}_{d_i} | \mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_{i-1}}) = \frac{pl(\mathbf{w}_{d_i} | \mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_{i-1}})^\gamma}{\sum_j pl(\mathbf{w}_{d_j} | \mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_{i-1}})^\gamma}$$

for factor based word probabilities

$$pl(\mathbf{w}_{d_i} | \mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_{i-1}}) = \frac{d(\hat{\mathbf{w}}_i, \sum_{j < i} \hat{\mathbf{w}}_j)}{\sum_{i'} d(\hat{\mathbf{w}}_{i'}, \sum_{j < i} \hat{\mathbf{w}}_j)},$$

with a distance measure d from a Euclidean space and the factor based feature representations of the Word-Vectors $\hat{\mathbf{w}}_j = \sum_{i=1}^T \omega_{d_j i} \mathbf{v}^i$. As proposed in [CJ98], we can use the cosine as similarity measure to define the final word probability:

$$pl(\mathbf{w}_{d_i} | \mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_{i-1}}) = \frac{\cos(\hat{\mathbf{w}}_i, \sum_{j < i} \hat{\mathbf{w}}_j)}{\sum_{i'} \cos(\hat{\mathbf{w}}_{i'}, \sum_{j < i} \hat{\mathbf{w}}_j)}.$$

Similar to the estimation of the likelihood for supervised topic models, supervised factors model like PLS can be similarly evaluated. We estimate the likelihood of document labels as

$$p(\mathbf{w}_{d_i}, \tau_d) = p(\mathbf{w}_{d_i})p(\tau_d | \mathbf{w}_{d_i}, \boldsymbol{\omega})$$

with

$$p(\tau_d | \mathbf{w}_{d_i}, \boldsymbol{\omega}) \propto e^{-\frac{1}{2} \|\tau_d - \boldsymbol{\omega}' \mathbf{w}_{d_i}\|^2}.$$

Since PLS uses linear regression for modeling the document features, we model the probability of a time stamp τ_d as proportional to the regression error of τ_d , under normality assumption with variance σ and mean $\boldsymbol{\omega}' \mathbf{w}_{d_i}$. This results in the following definition of the probability of a document d_i as Word-Vector \mathbf{w}_{d_i} given its latent factor representation $\hat{\mathbf{w}}_{d_i}$, all previous documents and a time stamp τ_d :

$$p(\mathbf{w}_{d_i} | \mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_{i-1}}, \tau_d) = \frac{pl(\mathbf{w}_{d_i} | \mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_{i-1}})^\gamma}{\sum_j pl(\mathbf{w}_{d_j} | \mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_{i-1}})^\gamma} \frac{e^{-\frac{1}{2} \|\tau_d - \boldsymbol{\omega}' \mathbf{w}_{d_i}\|^2}}{\sigma \sqrt{2\pi}}.$$

Using the last equation in the Sequential Monte Carlo model, we can estimate the likelihood of a given document with a time stamp for a supervised factor model.



4. Regularized Latent Variable Models

So far, the documents are represented as Word-Vectors or random sequences of words that are generated by a certain process. The approximation of this generation process of the documents by latent variables is done by finding optimal latent concepts of the corpus. The concepts are represented as latent factors in the VSM and as latent topics in the MM.

For factor models, a linear combination of latent factors (represented as vectors) is extracted such that the reconstruction error is minimized. The reconstruction error is the distance between the document in the vector space and the linear combination of latent factors that approximates it. For example, given a document as Word-Vector \mathbf{w}_d and its representation in terms of latent factors $\sum_i \omega_{di} \mathbf{v}^i$, the reconstruction error is

$$\|\mathbf{w}_d - \sum_i \omega_{di} \mathbf{v}^i\|_2^2,$$

the Euclidean distance between the two vectors. Each of the considered factor models can be

4. Regularized Latent Variable Models

described as extracting factors that minimize the reconstruction error.

In LSA, we minimize the reconstruction error

$$\|X - LER\|_2^2,$$

for the Term-Document Matrix and the matrices L, E and R . In NNMF, we minimize the reconstruction error

$$\|X - WV\|_2^2,$$

for the Term-Document Matrix and the two non-negative matrices W and V . In kPCA, we minimize the reconstruction error

$$\|\Phi(X) - P_U\Phi(X)\|_H^2,$$

for the mapping $\Phi(X)$ of the Word-Vectors into an RKHS and $P_U\Phi(X)$ the orthogonal projection of $\Phi(X)$ onto the subspace spanned by the principal functions of the covariance operator $C = \Phi(X)\Phi(X)'$. This results in a new kernel $\hat{K} = (P_U\Phi(X))'P_U\Phi(X)$.

For topic models, a number of latent random variables are extracted such that the likelihood of the corpus modeled by using these variables is maximized. The likelihood is the probability of the documents given the approximated generation process using the latent random variables. To avoid small probabilities that might result in mathematical overflows, the log-likelihood can be used. This is the logarithm of the likelihood.

In pLSA, we maximize the likelihood of the documents given the topics

$$\prod_d \prod_{w_n \in \mathbf{d}} p(d, w_n) = \prod_d \prod_{w_n \in \mathbf{d}} \sum_t p(d, w_n | t) p(t),$$

for the decomposition of the joint probability of the documents and words $p(d, w_n)$ over the latent variables t . In LDA, we maximize the likelihood of the documents given the topics

$$p(d | \alpha, \eta) = p(\mathbf{d} | \alpha, \eta) = \int \sum_t p(\theta, \beta, t, \mathbf{d} | \alpha, \beta) d\theta d\beta,$$

for a sequence of words \mathbf{d} in document d . The document probability $p(\mathbf{d} | \alpha, \eta)$ is modeled as marginal distribution of the latent topics t and the parameters θ and β for the document-topic distribution $p(d | t)$ with Dirichlet prior $\text{Dir}(\alpha)$, respectively the topic-word distribution $p(t | w)$ with Dirichlet prior $\text{Dir}(\eta)$.

4.1. Overview

The latent variable models described so far can be easily summarized as an optimization problem over latent variables that are specified by parameters Θ :

$$\Theta^* = \arg \text{opt}_{\Theta} L(\Theta),$$

for a loss function L and the optimal parameters Θ^* of either a minimization or maximization problem $\arg \text{opt}$. For the latent factor models, the loss function is the reconstruction error and

	LSA	NNMF	pLSA	LDA	kPCA
Θ	(E, L, R)	$(W, V \geq 0)$	$(p(w t), p(d t))$	(β, θ)	\hat{K}
L	$\ X - LER\ _2^2$	$\ X - WV\ _2^2$	$\prod_d p(d t)$	$\prod_d p(d t)$	$\ \Phi(X) - P_U \Phi(X)\ _H^2$
opt	min	min	max	max	min

Table 4.1.: Parameters for the latent variable models.

the optimization is a minimization. For the latent topic models, the loss function is the log-likelihood and the optimization is a maximization. The parameters Θ depend on the specific latent variable model. In Table 4.1, we summarize the loss functions, the optimizations and the parameters to be estimated for the different models.

In corpus linguistics, we consider several language resources with different corpora. The corpora contain documents from different times and from different sources. In diachronic and variety linguistics we use additional information about the documents like time stamps, the source or the genre of the document. For diachronic linguistics, we investigate the temporal distributions of certain concepts in the corpus. For variety linguistics, we investigate the distribution of the concepts across the sources and the genres of the documents.

To solve these linguistic tasks on large text corpora in heterogeneous language resources, we propose regularized versions of the factor and topic models. By regularization, we mean that the models are restricted in the following sense: the parameters of the model do not only optimize the reconstruction error or the likelihood of the documents, but also explain the additionally given information from the language resource. This information can be corpus specific or corpus unspecific. Corpus specific information are, for instance, document time stamps or genre information. Corpus unspecific information, on the other hand, can be information about words from dictionaries or WordNets and are valid for all corpora.

We assume to have additional information for the document as features \mathbf{x}_d and additional information about the words as features \mathbf{x}_w . To include these information into the latent variable models, we propose to add additional regularization terms $R(\Theta)$ into the optimization:

$$\Theta^* = \arg \text{opt}_{\Theta} L(X, \Theta) + R(\Theta).$$

The following sections show how we can use regularized models to leverage additional information from modern language resources to enrich corpora for analysis. These methods enable the integration of temporal information for diachronic linguistics and source or genre information for variety linguistics.

4.2. Topic Models with Regularization

In terms of topic models, we have a Bayesian model. In LDA, the regularization does make sense, since we can interpret priors and joint probabilities under appropriate independence assumptions as regularization terms of the log-likelihood. The dependency of the features is modeled as dependency of random variables. We distinguish three approaches to regularize topic models. First, so called upstream regularization includes regularization by modeling latent variables as depending on the additional information. In Figure 4.1 this is illustrated as a graph.

4. Regularized Latent Variable Models

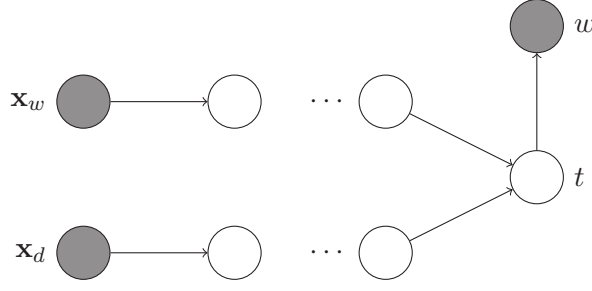


Figure 4.1.: Upstream Model in the Plate notation: Additional observed information about the documents and the words are integrated into the latent variable models by making the latent variables depending on them.

The latent variable t is dependent on information about documents \mathbf{x}_d and information about words \mathbf{x}_w . This dependence does not have to be direct. As in Dirichlet Multinomial Regression (DMR) for instance, the dependence is given by a prior with a meta parameter that depends on document features. A special Dirichlet prior is used on the parameter θ such that $\theta \sim \text{Dir}(e^{\mathbf{a}'_d \mathbf{x}_d})$ with $\mathbf{a}_d = [\mathbf{a}_1, \dots, \mathbf{a}_T]$. This results in the log-likelihood

$$L = L_1 + R,$$

for

$$L_1 = \sum_d \left(\frac{\Gamma(\sum_t \alpha_{dt})}{\Gamma(\sum_t \alpha_{dt} + n_d)} + \sum_t \frac{\Gamma(\alpha_{dt} + n_{d,t})}{\Gamma(\alpha_{dt})} \right. \\ \left. + \sum_t \frac{\Gamma(\sum_v \eta_{t,v})}{\Gamma(\sum_v \eta_{t,v} + n_k)} + \sum_v \frac{\Gamma(\eta_{t,v} + n_{t,v})}{\Gamma(\eta_{t,v})} \right)$$

and

$$R = \sum_t \log p(\mathbf{a}_{dt}) = \sum_t \log \text{N}(\mu_t, \sigma_t)$$

with $\alpha_{dt} = e^{\mathbf{a}'_{dt} \mathbf{x}_d}$. The loss is the log-likelihood of the collapsed likelihood of standard LDA:

$$L_1 = \log \prod_d p(\mathbf{d} | \alpha, \eta) = \log \prod_d \int_{\theta, \beta} p(\mathbf{d} | \alpha, \eta, \theta, \beta) p(\theta | \alpha) p(\beta | \eta) d\theta d\beta.$$

The regularizer R adds a regularization term that stems from the prior $p(\mathbf{a}_d)$. The parameters for the optimization are $\Theta = [\beta, \theta, \mathbf{a}_d]$.

Second, so called downstream regularization regularizes the topics to given document and word information by explicitly making this information depending on the latent variables. As seen in Figure 4.2 the given document and word information \mathbf{x}_d and \mathbf{x}_w depend on the latent variables t . The document and word features depend on the latent variables by joint probabilities. Hence, downstream regularization regularizes the latent factors, respectively the latent topics,

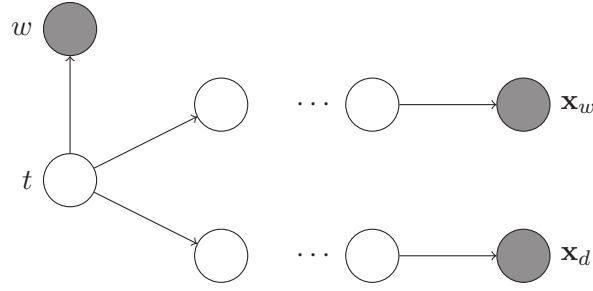


Figure 4.2.: Downstream Model in the Plate notation: Additional observed information about the documents and the words are integrated into the latent variable models by making them depending on the latent variables.

such that these information are modeled as depending on them (see Figure 4.2). An example for downstream regularized topic models is supervised LDA. In sLDA the dependency is given by the joint probability: $p(\theta, \beta, \mathbf{d}, \mathbf{t}, \mathbf{x}_d | \alpha, \eta)$, resulting in additional regularization terms in the log-likelihood

$$L = lB + \mathbb{E}_q[\log p(\mathbf{x}_d | \mathbf{t}, \mu, \sigma)],$$

for the lower bound (see Section 2.3.2)

$$lB = \mathbb{E}_q[\log p(\theta, \beta, \mathbf{d}, \mathbf{t} | \alpha, \eta)] - \mathbb{E}_q[\log q(\theta, \beta, \mathbf{t} | \gamma, \lambda, \phi)],$$

minimized by variational inference and the logarithm of the expectation of the document features given the topics with respect to the variational distribution q . Using collapsed Gibbs sampling (see Section 2.3.2) for sLDA, we minimize the likelihood

$$L = L_1 + R = L_1 + \log p(\mathbf{x}_d | \mathbf{t}, \mu, \sigma),$$

for the collapsed likelihood L_1 and the document features probability. The parameters for the optimization are $\Theta = [\beta, \theta, \sigma, \mu]$.

Third, so called off-stream regularization regularizes the latent variables indirectly by making the representation of the latent variables topic-word distribution depending on external information about the words. Figure 4.3 shows this as additional parallel path in the Plate notation of the latent variable model.

An example for off-stream regularized topic models is the method proposed by Petterson et al. [PSC⁺10]. The LDA parameter β is equipped with a Dirichlet prior that depends on word features by a functional relation f such that $\beta \sim \text{Dir}(f(\mathbf{x}_w))$. Hence, in terms of standard LDA we define $\eta = f(\mathbf{x}_d)$ and an additional prior is put on f . This results in adding a regularization term to the collapsed log-likelihood:

$$L = L_1 + R = L_1 + \log p(f, X_w).$$

Petterson et al. use $f = e^{a_{w,t} + a_w}$ independent of the word features \mathbf{x}_w but add an additional prior

$$R = \log p(f, X_w) \propto \sum_{\text{sim}(w,w')} \sum_t (a_{w,t} - a_{w',t})^2 + \sum_w a_w^2,$$

4. Regularized Latent Variable Models

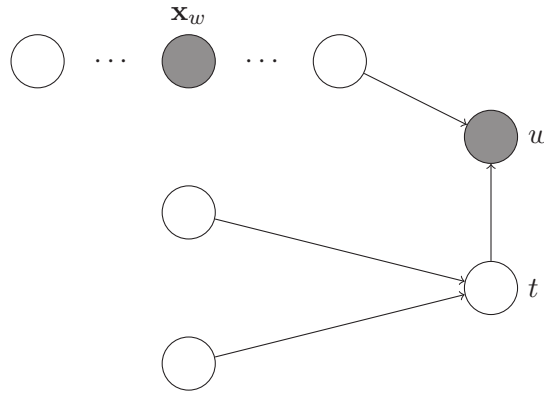


Figure 4.3.: Off-stream Model in the Plate notation: Additional observed information about the words are integrated into the latent variable models by making the only the word distribution depending on them.

for word features X_w indicating similarity between words with the relation `sim`. The parameters for the optimization are $\Theta = [\beta, \theta, a_w, a_{w,t}]$.

The notation of upstream and downstream regularization comes from previous works about how to integrate document features and labels into LDA. Mimno et al. [MM12] for instance describe DMR as upstream topic model in contrast to sLDA by Blei et al. [MB08] which is interpreted as downstream topic model. The notation off-stream regularization comes from the independence between the latent variables and the word information. The regularization is done purely by restricting the factors or the word distributions representing the latent variables. LDA with word features as proposed by [PSC⁺10] can be for instance interpreted as off-stream regularization.

4.3. Factor Models with Regularization

For factor models the notion of regularization is straightforward. As seen above, all factor models can be formulated as the optimization problem

$$\min_{\Theta} L(\Theta, X),$$

for a factorization of the Term-Document Matrix X , the parameters of this factorization Θ and a corresponding loss function L . In NNMF the factorization is $X = WV$ for example and in LSA we have $X = LER$. Restricting the resulting factors towards given document and word information is done by putting a regularization term R into the optimization problem that punishes factors that do not go along with these information. This results in the new optimization problem:

$$\min_{\Theta} \lambda_1 L(\Theta, X) + \lambda_2 R(\Theta, X_d, X_w).$$

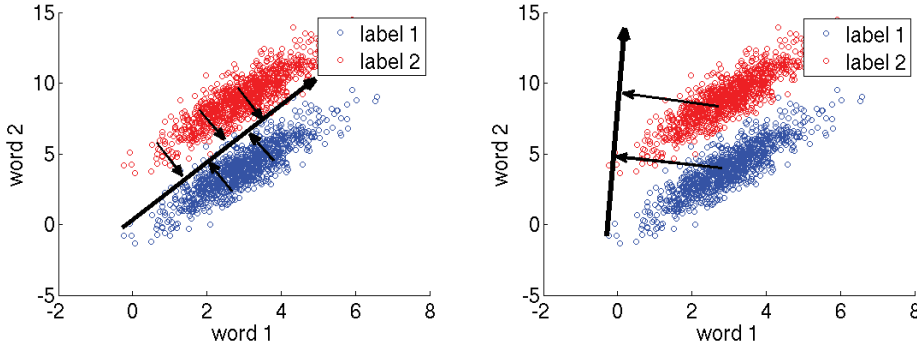


Figure 4.4.: Illustration of regularization of factors for given documents with labels 1 and 2 for two words in the VSM. We plot the Word-Vectors of documents with label 1 as blue dots, documents with label 2 as red dots. The axis mark the number of occurrences of the words. Left: Projecting data onto possible factors with low reconstruction error, but with no correspondence to given labels. Right: Projection onto possible factors that can discriminate documents with different labels.

The weights λ_1 and λ_2 account for different emphasizes on the reconstruction error or the modeling quality of the factors for the given document or word features. Larger λ_1 result in factors that might better represent the Word-Vectors but have lower correspondence to the document and word information. Larger λ_2 on the other hand result in better correspondence to the document and word information but at the expense of possible worse representations of the Word-Vectors.

We distinguish two approaches for regularization in factor models. First, norm or regression based regularization adds a term to the optimization problem that estimates the distance of a factor matrix and a regularization matrix. The factor matrix contains the factors as row or column vectors. The regularization matrix contains document and word features. This can be interpreted as downstream regularization since the document and word features depend on the latent factors. Second, subspace and projection based regularization restricts the subspace respectively the projection onto the subspace spanned by the factors. This can be interpreted as upstream regularization since the factors depend on the document or word features.

In norm based regularization, the additional information from the language resources is modeled as feature matrix X_d consisting of the document feature vectors \mathbf{x}_d and feature matrix X_w consisting of the word feature vectors \mathbf{x}_w . The dependency of the features is modeled by distances between latent factors and document features, respectively word features.

In LSA for example, the optimization problem can be formalized as regularized version

$$\min_{L,E,R} \lambda_1 \|X - LER\|_2^2 + \lambda_2 R(\Theta, X_d, X_w),$$

with

$$R(\Theta, X_d, X_w) = \lambda_2 \frac{1}{2} \|X_d - AL\|_2^2 + \lambda_3 \frac{1}{2} \|X_w - BR\|_2^2,$$

for $\Theta = [E, L, R, A, B]$. This regularization models the features as linear regression of the factors with parameter matrices A and B .

4. Regularized Latent Variable Models

To solve the optimization problem we need to additionally optimize over the parameter matrices A and B . This can be done in an alternating manner. First, we find the optimal factorization of the term-document matrix for the optimization problem given A and B . Next, we find the optimal parameter matrices by minimizing R via gradient descent. The gradients are

$$\frac{\partial \frac{1}{2} \|X_d - AL\|_2^2}{\partial A} = -A' \|X_d - AL\|_2$$

and

$$\frac{\partial \frac{1}{2} \|X_w - BR\|_2^2}{\partial B} = -B' \|X_w - BR\|_2.$$

This regularizer results in factors that correspond to the document and word information with an assumed linear relation. For example, given two possible labels for the documents, this regularization shall influence the factors such that they reflect these labels. In Figure 4.4, we illustrate this for documents that contain only two words and additional document labels 1 and 2. The labels could be, for instance, indications of positivity or negativity of the texts. While standard LSA would, for instance, result in factors that approximate the Word-Vector well, a regularized LSA can be used to punish factors that do not separate the documents by the labels. To solve the diachronic linguistic task as described above, we define the matrix $X_d = T$ of document time stamps. The vector T contains at component i the time stamp τ_i from document d_i . This approach has the very strong assumption that the time stamps depend linearly on the factors.

Similar to modeling the document and word features as results of a regression based on factors, we can also model the factors as results of a regression based on document and word features. This can be formulated using the regularization term

$$R(\Theta, X_d, X_w) = \lambda_1 \|L - A'X_d\|_2^2 + \lambda_2 \|R - B'X_w\|_2^2,$$

with $\Theta = [E, L, R, A, B]$.

In contrast to regression based regularized models, purely norm based regularized models regularize the latent factors such that the matrix of the factors have low distance to the matrix containing information from document and word features. An example for norm based regularized factor models is the model:

$$\min_{L, E, R} \lambda_1 \|X - LER\|_2^2 + \lambda_2 \|L - X_d\|_2^2 + \lambda_3 \|R - X_w\|_2^2,$$

with $\Theta = [E, L, R]$. Here, we look for a low-dimensional feature representation of the Word-Vectors with smallest reconstruction error and matrices L, R containing factors that are similar to given regularization matrices X_d and X_w . The regularization matrix X_w for instance can contain prior information about the words. In variety linguistic tasks, this could be used to force the word distribution in the factors to match a given word distribution from a different data set. This approach can be interpreted as off-stream regularization since the document and word features influence the factors only indirect.

In subspace based regularization, we model the words, the latent factors and the additional information about the documents and the words jointly. Jointly modeling latent factors with

4.3. Factor Models with Regularization

document and word features can be done by aligning the subspaces that are spanned by the latent factors with the subspaces spanned by the document features for instance. This regularizes the factors based on the subspace respectively the projection onto the subspace spanned by the factors.

Starting with LSA, we are minimizing $\|X - LER\|_2^2$ such that $L'L = I$ and $R'R = I$. The constraints $L'L = I$ and $R'R = I$ make them orthonormal and the row respectively the column vectors are basis vectors of the space spanned by the Word-Vectors, respectively the Document-Vectors. If we are considering only document features, we need only the right singular vectors. The subspace that is spanned by the right singular vectors is such, that the projection of the Word-Vector via projection matrix $P = RR'$ onto this space results in the smallest 2-norm. This can be easily re-formulated as the optimization problem

$$\text{opt}_{\Theta} L(X, \Theta) = \max_{P: I=P'P} \|PX\|_2^2,$$

for any projection matrix P . Hence, we optimize over $\Theta = P$.

In order to align the subspace with the feature space from the document features, we add a regularization term that penalizes P when the subspace $\text{span}(P)$ has large principal angles to the feature space from the document features.

This approach uses the interpretation of the factors as basis of a subspace in the space spanned by the Word-Vectors. We can further interpret the document features as drawn from a feature space. Aligning subspaces between the factor space and feature space means we maximize covariance between the corresponding bases. We assume that the factors span a subspace in the same $\mathbb{R}^{\max(p,V)}$ as the features from the documents. This is possible since the Word-Vectors and the factors are vectors in \mathbb{R}^V . The overall optimization problem with subspace based regularization to document features is

$$\max_{\Theta=[P:I=P'P]} \lambda_1 \|P'X\|_2^2 - \lambda_2 R(\Theta, X_d).$$

The first part results in a low dimensional feature representation of the Word-Vectors as linear combination of factors as in LSA. The second part regularizes the factors with respect to the document features. Here, we concentrate on linear factor models and only document features like time stamps. It is straightforward to extend the proposed model to word features. Later, we will also discuss non-linear factor models by kernel methods.

Similar to regression based regularization, we use subspace regularization to align the subspace spanned by the factors to document labels like time stamps. Since we are only interested in the factors for the Word-Vectors, we concentrate on regularizers in the form of

$$R(\Theta, X_d) = \text{tr } P'MP, \text{ s.t. } P'P = I,$$

for the matrix $M = X'X_dX_d'X'$. This regularizer has its maximum at the matrix P projecting onto eigenvectors of the matrix M . These eigenvectors are the principal vectors of the subspaces spanned by the document features. This means, projecting the Word-Vectors via P results in a new word representation in a subspace that maximally aligns with the document features. In Figure 4.5, we visualize this on a low dimensional example for documents containing only two words with time stamps.

4. Regularized Latent Variable Models

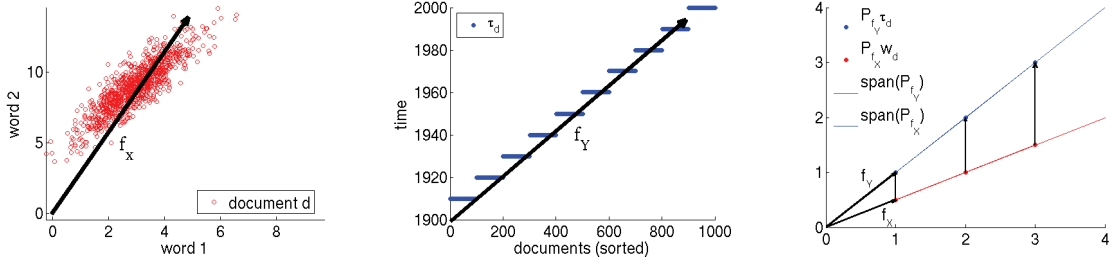


Figure 4.5.: Visualization of subspace based regularization by maximizing the correlation between Word-Vectors and time stamps. Left: Word-Vectors for a corpus in the VSM with number of word occurrences marked on the axis and a latent factor as extracted by LSA. Middle: Time stamps for the documents from the corpus plotted against the number of documents in some order of the documents and the time in years from 1900 to 2000. Right: In an ambient space containing both factors (from the Word-Vector and the time), projecting the time stamps via P_{f_Y} and the Word-Vector via P_{f_X} onto the corresponding subspaces $\text{span}(P_{f_Y})$ and $\text{span}(P_{f_X})$ results in a maximal correlation between the time and documents.

Besides this subspace regularization, we can also regularize the projection onto this subspace. This means we restrict the regions where the projection matrix P projects the Word-Vector with respect to given document and word information. Given, for example, corpus unspecific information about words from a WordNet [Mil95], we want the latent variables to reflect these relations in the generation of the documents. If we know, for instance, that two words are highly similar based on WordNet, these words shall also be similar in terms of the latent factors, respectively latent topics. For latent factors models, these words shall have low distance in the subspace spanned by the factors. This means, projecting the two Word-Vectors that represent the words (hence sparse vectors containing only one non-zero entry at the component corresponding to the word) onto this subspace shall decrease their Euclidean distance.

In Figure 4.6, we illustrate this regularization. The Word-Vectors w^1 , w^2 and w^3 represent three similar words w_1 , w_2 and w_3 based on WordNet. The similarity is quantified by a function $\text{sim}(w_i, w_j)$, measuring the strength of the similarity. For WordNet, this similarity measure can be for instance the distance in the WordNet graph. The regularization shall restrict the factors such that a projection onto the subspace which is spanned by the factors mapped similar words close together. A similar approach can be used to regularize the factors such that similar documents are projected close to each other in the subspace spanned by the factors. Such a similarity can be given by document labels, time stamps or user specific information. This can be used to make whole sets of documents more similar on the subspace spanned by the factors.

The regularization terms to perform a projection based regularization punishes projections that result in large distance of words or documents from which we have additional information about their similarity. To force similar words to be projected close to each other in the space

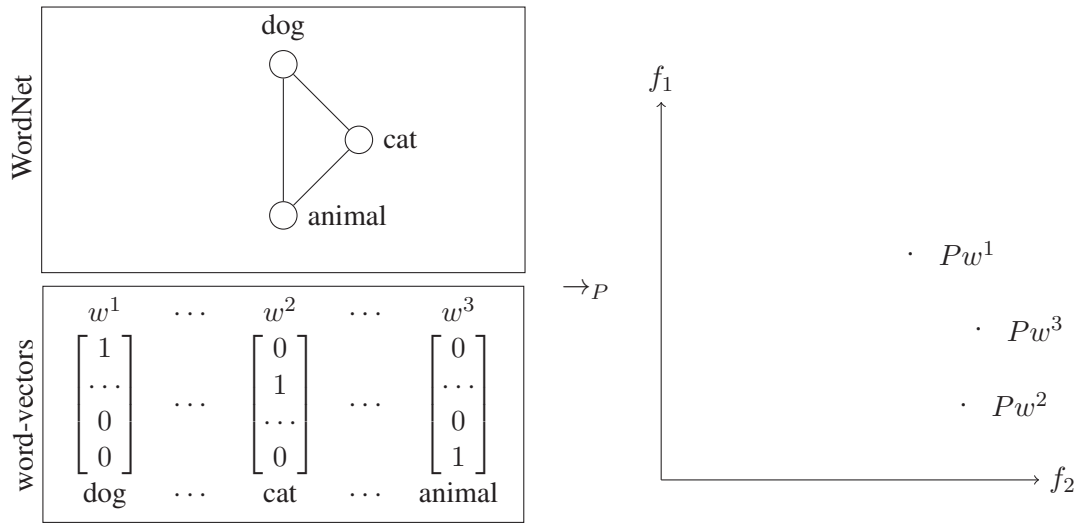


Figure 4.6.: Illustration of jointly modelling words and similarity information about the words in a factor model. Given similarity information about words from WordNet like the close semantic relation of the words *dog*, *cat* and *animal*, this similarity shall also be reflected in the subspace spanned by the factors. The words represented as Word-Vector are projected such that the Euclidean distance in the subspace is small.

spanned by the factors, we use the regularization term

$$R(\Theta, X_w) = \sum_{\text{sim}(w_i, w_j)} \|P\mathbf{w}^i - P\mathbf{w}^j\|_2^2,$$

for X_w word features containing similarity information from WordNet for example. To force Word-Vectors of similar documents to be close in the latent subspace, we use

$$R(\Theta, X_d) = \sum_{\text{sim}(d_i, d_j)} \|P\mathbf{w}_{d_i} - P\mathbf{w}_{d_j}\|_2^2,$$

for X_d document features containing similarity information about the documents.



5. Use Case Diachronic Linguistics

In a first use case, we will introduce a regularized topic model that efficiently integrates temporal information. In diachronic linguistics, we have additional information about the time of the documents, telling us when the text was composed. This information can be used to investigate the distribution of certain linguistic phenomena over time. We develop a regularized topic model that can use time stamps as document labels. The main contribution is the introduction of an attention based regularization of a topic model. This attention model coincides with observations from linguistic research of the temporal distribution of certain linguistic phenomena.

5.1. Motivation

With the availability of corpora of large text collections over a long period of time, empirical analyses of the temporal distribution of the texts are possible. For example, the frequency of

5. Use Case Diachronic Linguistics

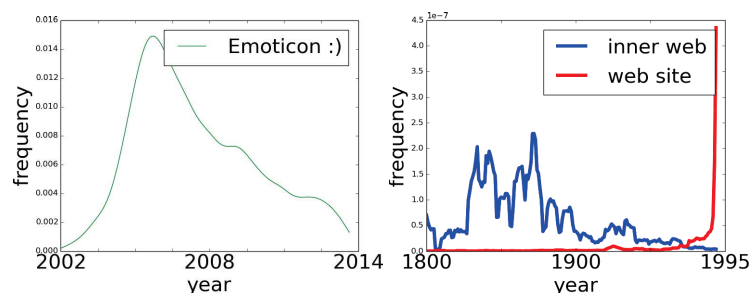


Figure 5.1.: Temporal distributions of tokens in different text collections. Left: Frequency (y-axis) of usage of smilies (:)) in Wikipedia talk pages from year 2002 to 2014 (x-axis). Right: Frequency (y-axis) of the word "web" in the meaning of cloth versus web page from Google n-grams from year 1800 to 1995 (x-axis).

the usage of Emoticons¹ in social media content over time can be used to analyze Internet-based communication. We assume that certain characteristics of the usage and the frequency have a clear temporal aspect. The frequency is not constant over time, but follows a law that is physically motivated. First, the usage of Emoticons gets hyped until a maximum is reached. After this maximum, we expect only decrease in frequency. The decrease smoothly reaches a saturation. In Figure 5.1 on the left, the frequency of the Emoticon ":" is plotted over time. We can apply this assumption on latent aspects of the texts. Given for instance the text collection of books used for the Google n-grams viewer. We expect that certain meanings in the text covered by the latent aspects in the documents follow a similar frequency in usage over time. Texts containing the word *web* for example, will more likely speak about cloth when they are written before 1980s. The word *web*, is additionally used as web page from that time on. In Figure 5.1 on the right, we show the corresponding frequency over time from the Google n-gram corpus.

Given time stamps for the documents in a given corpus, we want that the latent variables from a topic model do not only explain how the words are generated but also how the time stamps is generated. Hence, we assume that the concepts in the documents also influence the time stamps. Some concepts will be only associated with certain times. This shall be reflected in the topics. For example, documents d_i with the same time stamp τ shall have the decomposition of the joint probability

$$\prod_d \prod_{w_n \in \mathbf{d}} p(d, w_n) = \prod_d \prod_{w_n \in \mathbf{d}} \sum_t p(d, w_n, \tau|t)p(t) = \prod_d \prod_{w_n \in \mathbf{d}} \sum_t p(d, w_n|t)p(\tau|t)p(t),$$

such that the probability $p(\tau|t)$ of the time stamp given a topic puts most of its probability mass on a small number of topics. Such joint models of documents and time can be used to perform diachronic linguistics tasks using corpora with temporal information. In Figure 5.2, we illustrate this for a topic model on Wikipedia talk pages containing the term *president*.

¹a pictorial representation of an emotion with ASCII symbols

5.2. Related Work in Temporal Topic Modeling

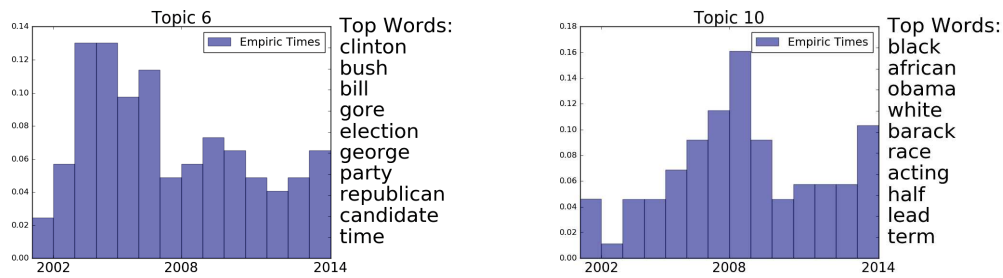


Figure 5.2.: Amount of usage (via probabilities marked on the left y-axis) of topics over time from year 2002 to 2014 (marked on the x-axis). The most important words for the topics are marked on the right y-axis with most important words at the top. Certain concepts in a corpus with additional time information have a clear temporal concentration. On the left, we see that *Bill Clinton* is used in Wikipedia discussions more prominently in the years before 2007, *Barack Obama* and his heritage is discussed mostly around 2008.

5.2. Related Work in Temporal Topic Modeling

There are different approaches to model time in topic models. Closest related to our approach is the work by [WM06]. The authors use a continuous non-Markovian approach to model time within topics. The main difference is the assumed distribution over time. While Wang and McCallum assume that the time stamps of a document are Beta distributed, we assume a physically motivated distribution that can model attentions. In our experiments this assumption results in more reasonable topics. In [WR12], the authors propose a nonparametric mixture model of time for topic models. This makes it necessary to use complicated Dirichlet Processes and restricts the used mixture of time distributions to have simple conjugates to perform efficient inference. That is why they use a mixture of Gaussians. We argue that Gaussians are not appropriate for modeling time since they are symmetric, which is unrealistic for time periods. On the other hand, enough components of Gaussians might be able to model time correctly. However, this makes the inference more complicated and the model will overfit easily. A similar approach has been proposed by [DHWX13]. The main difference to [WR12] is that the authors use an additional Hierarchical Dirichlet Process prior to remove the assumption of a fixed given number of topics.

All these approaches model the documents, respectively the words, and the time jointly. Further approaches to model topics and time are sequential, using for instance, Markov chains. Such sequential models estimate conditional (transition) probabilities of a topic at time τ_i given the topic at time τ_{i-1} . While standard Hidden Markov Models can handle such sequences of latent topics, they do not cover a sequence of word-distributions associated with each topic as in LDA. One way to compensate for this is to explicitly model sequences of multinomial distributions with Dirichlet priors drawn from a Dirichlet Processes. As proposed by [SR05] the Dirichlet Process can be defined as depending on additional information like times associated with observed documents. Dynamic topic models as proposed in [BL06b] for discrete time and

5. Use Case Diachronic Linguistics

in [WBH12] for continuous time model topics as sequences over time using state space models with Normally distributed transition probabilities of topic-word distributions. In [AX12] a hierarchical Dirichlet process is used to model a possibly infinite number of topics. These approaches model the evolution of topics. They can not model the temporal distribution of a topic.

Also standard LDA can be used to investigate temporal behavior. Based on post-processing of results from LDA, [NSS11] estimate trends and evolutions of topics. Such post-hoc approaches have also been used to investigate topics over time in [HJM08]. Further work on temporal topic modeling concentrates on the visualization. In [PZS⁺13], the authors propose a clustering method to group texts and segment these groups for visualization over time. [GJG⁺15] estimate a segmentation over time an apply standard LDA on windows of texts from these segments.

5.3. Topic Models for Diachronic Linguistic Tasks

For topic models like LDA, a regularization with respect to given time stamps can be derived starting with different versions of supervised topic models. For example supervised LDA (sLDA) [MB08] or Topics over Time (TOT) [WM06] model the regularization of the topics to external information about documents by joint probabilities. The document labels are modeled as observed random variable and a joint model is estimated.

We concentrate on the collapsed version of supervised topic models in the following sense: Instead of observing a label (the document information) for each document, we assume that we observe this label for each token in the document. Under this assumption, we can easily use a collapsed Gibbs sampler for inference. In Figure 5.3, we show these different approaches graphically. As argued by Mimno et al. in [WM06], the collapsed version of sLDA results in a collapsed Gibbs sampler

$$p(t_i|w, l) \propto \frac{n_{w,t_i} - 1 + \eta}{n_{t_i} - 1 + W\eta} \cdot (n_{d,t_i} + \alpha)p(l),$$

for topic t_i given word w and a document label l for document d .

The first part of the right hand side of the last equation comes from standard LDA Gibbs sampler (cf. Section 2.3.2) and the second part is the density of the labels l . Besides simpler inference, we can also use this approach to include either document features or word features. A disadvantage is that we cannot include structured information about words like word groups or correlations due to the independence assumptions. Later, we will explain how we use upstream regularized topic models to include correlation information about words in LDA in an additional use case.

Next, we explain how temporal topic models solve the diachronic linguistic tasks by downstream regularization. In this scenario, we assume the document labels l to be time stamps associated with each document.

5.3.1. Temporal Topic Modeling

While the standard topic models group only words and documents in semantically related topics (cf. Section 2.3.2), we are further interested in the distribution of the topics over time. Certain

5.3. Topic Models for Diachronic Linguistic Tasks

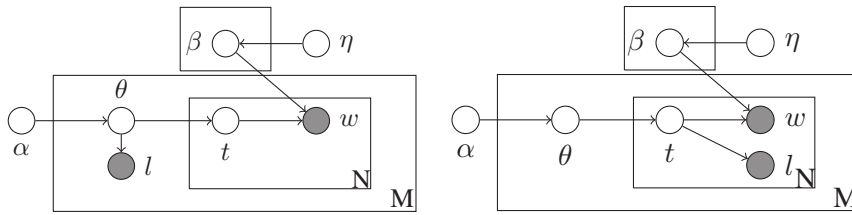


Figure 5.3.: Left: sLDA represented as graphical model in the Plate notation. In contrast to LDA, we additionally model a label as observed random variable that depends on the topic distribution. Right: Collapsed sLDA represented as graphical model in the Plate notation. Labels are modeled as observed random variables that depend on each topic.

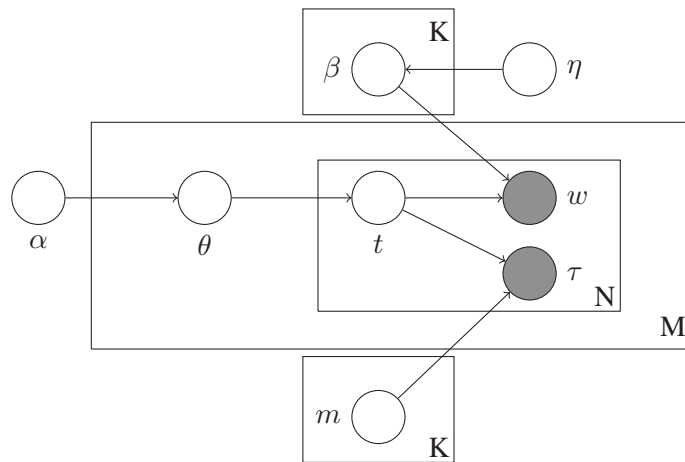


Figure 5.4.: Temporal LDA represented as graphical model in the Plate notation. As in the collapsed version of supervised LDA, the time stamps are modeled as observed random variables that depend on each topic.

meanings of words might be used only in certain time periods. The word *cloud*, for instance, has recently become a new meaning of data cloud. Further, there can be certain trends or attentions to topics. Topics about US presidents, for example, will very likely be highly present around a year of elections.

In order to extract the distribution of topics over time, we use topic models that consider temporal information about the documents. Each document has a time stamp τ . As described above, we further assume that each token in the documents is associated with this time stamp. The time stamps follow the distribution $p(\tau)$. The time stamps are conditionally independent given a topic. This means, given a sequence of topic assignments $\mathbf{t} = (t_1, \dots, t_N)$ for word tokens w_1, \dots, w_N from a corpus and associated time stamps $\boldsymbol{\tau} = (\tau_1, \dots, \tau_N)$, the probability

5. Use Case Diachronic Linguistics

of the time stamps is

$$p(\boldsymbol{\tau}|\mathbf{t}) = \prod_n^N p(\tau_n|t_n).$$

Just as sLDA, the integration of the time stamps into LDA is done by modeling the time stamps as additional observed random variable that depend on the topics. In Figure 5.4, the graphical representation of LDA with time is depicted. The joint probability is

$$p(\mathbf{t}, \mathbf{d}, \boldsymbol{\tau}|\alpha, \eta, m) = p(\mathbf{t}|\alpha)p(\mathbf{w}|\mathbf{t}, \beta)p(\boldsymbol{\tau}|\mathbf{t}).$$

It is straight forward to define a Gibbs sampler for this distribution. Analogue to LDA with Gibbs sampling, we sample from

$$p(t_i|\mathbf{t}^{-i}, w_i, \boldsymbol{\tau}) = \frac{p(\mathbf{t}_{+t_i}^{-i}, w_i, \boldsymbol{\tau})}{p(\mathbf{t}^{-i}, w_i, \boldsymbol{\tau})} \quad (5.1)$$

$$\begin{aligned} &= \frac{p(\mathbf{t}_{+t_i}^{-i}, w_i) \prod_n p(\tau_n|t_n)}{p(\mathbf{t}^{-i}, w_i) \prod_{n \neq i} p(\tau_n|t_n)} \\ &\propto p(\mathbf{t}, w_i)p(\tau_i|t_i). \end{aligned} \quad (5.2)$$

Topic Models over Time

A specific instance of the probability of the time stamps is the Beta distribution. Wang and McCallum [WM06] introduced this model to investigate topics over time. They call this method: TOT. The generative process of standard LDA is extended such that for each word w_i in each document, we also draw a time stamp $\tau_i \sim \text{Beta}(a, b)$ with (a, b) the shape parameters of the Beta distribution.

The shape parameters are estimated by the method of moments. After each Gibbs iteration the parameters are estimated in the following way: For each topic t we estimate the empirical mean \hat{m} and sample variance s^2 of all time stamps from the documents that have been assigned to this topic. By the method of moments, we set $a = \hat{m} \cdot (\frac{\hat{m} \cdot (1 - \hat{m})}{s^2} - 1)$ and $b = (1 - \hat{m}) \cdot (\frac{\hat{m} \cdot (1 - \hat{m})}{s^2} - 1)$ for each topic. Integrating the time stamp as Beta distributed random variable, we the probability of a topic t_i , given a word w in a document d with time stamp τ_i and all other topic assignments

$$p(t_i|w, \tau_i, \mathbf{t}^{-i}) \propto \frac{n_{w,t_i} - 1 + \eta}{n_{t_i} - 1 + W \cdot \eta} \cdot (n_{d,t_i} + \alpha) \cdot \frac{(1 - \tau_i)^{a_i-1} \cdot \tau_i^{b_i-1}}{B(a_i, b_i)}, \quad (5.3)$$

where the last term originates from the density of the Beta distribution

$$\text{Beta}(\tau; a, b) = \frac{\tau^{a-1}(1 - \tau)^{b-1}}{B(a, b)},$$

at time stamp τ and $B(a, b)$ the Beta function.

In contrast to the approach by Wang and McCallum, we use Maximum Likelihood Estimation to find the optimal parameters for the Beta distribution for each topic. This is more consistent

5.3. Topic Models for Diachronic Linguistic Tasks

with the model estimation for LDA with Gibbs sampling. To ensure the positivity of the parameters for the Beta distribution, we redefine the Beta distribution as $\text{Beta}(e^a, e^b)$. This leads to the following log-likelihood for a sequence of time stamps $\boldsymbol{\tau} = (\tau_1, \dots, \tau_N)$:

$$L(\{\tau_n\}_{n=1}^N, a, b) = (e^a - 1) \sum_n \tau_n + (e^b - 1) \sum_n \log(1 - \tau_n) - N \log(\text{B}(e^a, e^b)).$$

The gradient $\nabla_{a,b}L$ of the parameters a and b of this log-likelihood is

$$\nabla_{a,b}L = \begin{pmatrix} \frac{\partial L}{\partial a} \\ \frac{\partial L}{\partial b} \end{pmatrix},$$

with

$$\frac{\partial L}{\partial a} = \frac{\partial L}{\partial e^a} \frac{\partial e^a}{\partial a} = \frac{\partial L}{\partial e^a} e^a,$$

respectively

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial e^b} \frac{\partial e^b}{\partial b} = \frac{\partial L}{\partial e^b} e^b.$$

The partial derivatives of the log-likelihood with respect to the transformed parameters are

$$\frac{\partial L}{\partial e^a} = \sum_n \log \tau_n - N(\Psi(e^a) - \Psi(e^a + e^b))$$

and

$$\frac{\partial L}{\partial e^b} = \sum_n \log(1 - \tau_n) - N(\Psi(e^b) - \Psi(e^a + e^b)).$$

Using the gradient information, we perform Newton-like gradient descent with a standard BFGS optimization solver [LN89] after each Gibbs sampling iteration.

Topics with Attention Curves (@TM)

Considering TOT, it is very unrealistic that topics appear suddenly and then vanish. Beta distributions model the time as sharp intervals as to be seen in Figure 5.5 on the right. We propose to use a physically motivated model that is able to express smoother declines and has been successfully applied to model attentions in social media [BHK15]. As proposed by Bauchhage and Kersting in [BK14], diffusion models like the Bass model [Bas69] can be used to model attentions. While Bauchhage and Kersting concentrate on search queries in the Internet, we apply this idea on topics over time. We use the Shifted-Gompertz distribution to model attentions on certain topics. The density of the Shifted-Gompertz distribution is

$$\text{SG}(\tau; a, b) = b e^{-b\tau} e^{-a e^{-b\tau}} (1 + a(1 - e^{-b\tau})),$$

with a scaling parameter b and a shape parameter a .

5. Use Case Diachronic Linguistics

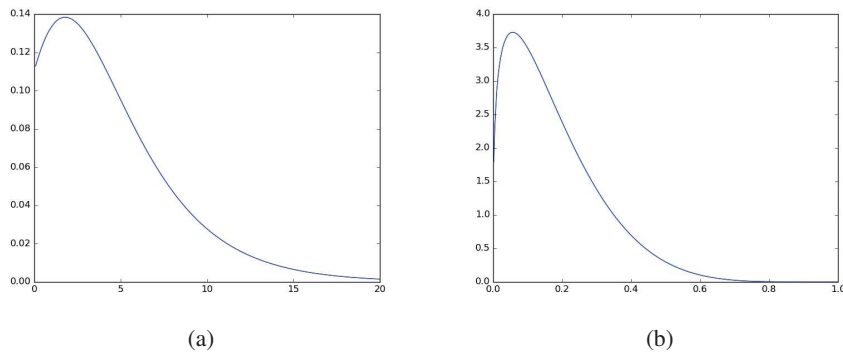


Figure 5.5.: Difference between the Shifted-Gompertz density (a) and the Beta distribution density (b): Gompertz decreases smoother and does not go to zero rapidly.

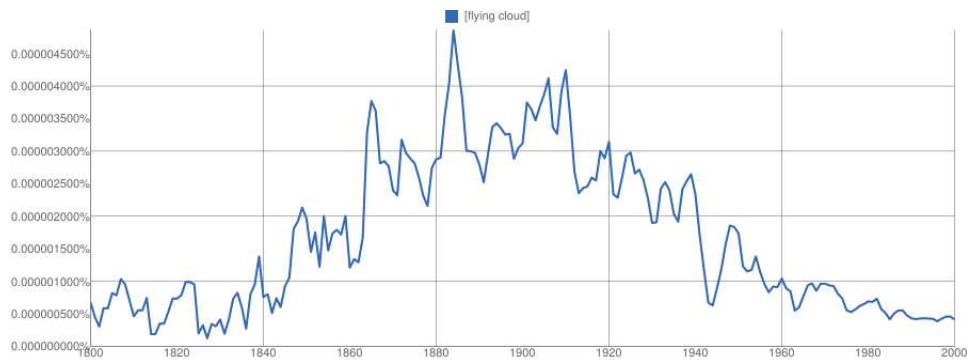


Figure 5.6.: Course of the frequency (y-axis) of the bi-gram "flying cloud" in texts from Google n-gram viewer from year 1800 to 2000 (x-axis).

In Figure 5.5 on the left for instance, we see that the Shifted-Gompertz still puts probability mass larger than zero for larger values (hence later times). This means that topics with attention curves never vanish completely. This makes sense, since we expect that certain topics might become less likely over time, but the probability that this topic will appear again is not zero.

Now the question is whether this physically motivated time model is also valid for the topics extracted by a topic model. We assume that the topics follow a growth and decline phase over time. Depending on the corpus, we might cover only certain periods of the phases. We illustrate this on possible word senses induced by topics from the corpus. For instance, the word *cloud* in the meaning of "computing and storage on demand" in a balanced Newspaper corpus from 1900 to 2014 is expected to be in the growth phase from the 70s to present. The decline phase has not started yet. On the other hand, we can expect the usage of the word *cloud* in the meaning of weather to be at a constant peak, that covers all the time. By contrast, the word *cloud* together with *flying* can be assigned to the meaning of ship *Flying Cloud*. Topics that cover this Named Entity have a high peak at the beginning of the 20th century and start to decline. This can be seen from the Google n-gram viewer as depicted in Figure 5.6. In the ideal case, the growth and

decline of a topic is completely covered in the corpus. Hence, from a corpus with more data from before 1900, we might find both phases of the topic from growth to decline.

Analogously to TOT, we sample the topics from the distribution

$$p(t_i|w, \boldsymbol{\tau}, \mathbf{t}^{-i}) \propto \frac{n_{w,t_i} - 1 + \eta}{n_{t_i} - 1 + W\eta} (n_{d,t_i} + \alpha) \text{SG}(\tau_i; a_i, b_i). \quad (5.4)$$

to estimate topics with attention curves. Throughout this thesis, we call the method Attentional Topic Model (@TM).

The parameters a and b are estimated by Maximum Likelihood Estimation after each Gibbs sampling iteration. To ensure positivity of the parameters, we redefine the Shifted-Gompertz to $\text{SG}(e^a, e^b)$ with transformed parameters e^a and e^b . The log-likelihood of N Shifted-Gompertz distributed time stamps $\boldsymbol{\tau} = (\tau_1, \dots, \tau_N)$ is

$$L(\{\tau_n\}_{n=1}^N, a, b) = Nb - e^b \sum_n \tau_n - e^a \sum_n e^{-e^b \tau_n} + \sum_n \log(1 + e^a(1 - e^{-e^b \tau_n})).$$

The gradient $\nabla_{a,b}L$ of the parameters a and b of this log-likelihood is

$$\nabla_{a,b}L = \begin{pmatrix} \frac{\partial L}{\partial a} \\ \frac{\partial L}{\partial b} \end{pmatrix},$$

with

$$\frac{\partial L}{\partial a} = \frac{\partial L}{\partial e^a} \frac{\partial e^a}{\partial a} = \frac{\partial L}{\partial e^a} e^a,$$

respectively

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial e^b} \frac{\partial e^b}{\partial b} = \frac{\partial L}{\partial e^b} e^b.$$

The partial derivatives of the log-likelihood with respect to the transformed parameters are

$$\frac{\partial L}{\partial e^a} = - \sum_n e^{-e^b \tau_n} + \sum_n \frac{1 - e^{(-e^b \tau_n)}}{1 + e^a(1 - e^{-e^b \tau_n})}$$

and

$$\frac{\partial L}{\partial e^b} = \frac{N}{e^b} - \sum_n (1 - e^a e^{-e^b \tau_n}) \tau_n + e^a \sum_n \frac{e^{-e^b \tau_n}}{1 + e^a(1 - e^{-e^b \tau_n})}.$$

Using these gradient information, we perform Newton-like BFGS for the Maximum Likelihood Estimation to find the optimal Shifted-Gompertz distribution for each topic.

Online Downstream Regularization

Analogously to online LDA, downstream regularized topic models like sLDA, Topics over Time topic models or topic models with attention curves can also be solved in an online manner. We concentrate on topic models with attention curves (@TM) to illustrate this. As in online

5. Use Case Diachronic Linguistics

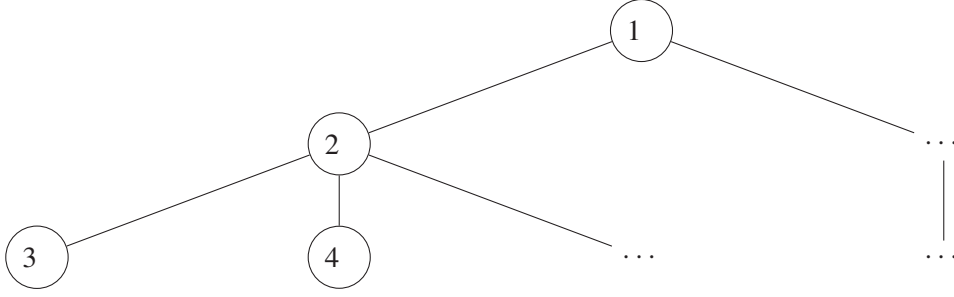


Figure 5.7.: Example of a topic hierarchy from a random tree based on samples from a nested Chinese Restaurant process.

LDA, we need to define a variational bound for the joint probability of a sequence of words $\mathbf{w} = (w_1, \dots, w_N)$ and a sequence of corresponding time stamps $\boldsymbol{\tau} = (\tau_1, \dots, \tau_N)$:

$$p(\mathbf{d}, \boldsymbol{\tau} | \alpha, \eta, m) \leq L_{\boldsymbol{\tau}}.$$

For the regularization we use the loss function

$$L_{\boldsymbol{\tau}} = L + \mathbb{E}_q[\log p(\boldsymbol{\tau} | \mathbf{t})]$$

as bound with L as defined in Equation 2.19 and

$$\mathbb{E}_q[\log p(\boldsymbol{\tau} | \mathbf{t})] = Nb - e^{bt} \sum_n \tau_n - e^{at} \sum_n e^{-e^{bt} \tau_n} + \sum_n \log(1 + e^{at}(1 - e^{-e^{bt} \tau_n})).$$

This is similar to the variational bound for sLDA. Consequently, we separate the parameters into global and local parameters as in online LDA. The global parameters are λ as in online LDA and additional a_t and b_t . The online algorithm for topic models with attention curves is the following: For each batch of documents, we estimate the local parameters as in online LDA. After this, we perform maximum likelihood estimations for the parameters λ , a_t and b_t .

5.3.2. Hierarchical LDA over Time

The topics extracted by TOT and @TM assume no further structure among the topics. This assumption is very weak since topics might be related. Especially over time, some topics might be periodic, reappearing or mixtures of smaller topics. The mentions of the president of the USA for example will appear every four years in news papers articles or social media content. Each appearance can be modeled by a topic with an attention curve. All these topics belong to the same more general topic *president*. We propose to model such topics in a hierarchical way.

The hierarchical model we use, is an extension of the hierarchical LDA by Blei et al. [BGJT04]. In contrast to their approach, we additionally model the time of topics with attention curves. Hierarchical LDA is non-parametric in the number of topics. We only specify the depth of the topic hierarchy. A fully-non-parametric model that estimates the best depth is also possible, but not considered in this work.

5.3. Topic Models for Diachronic Linguistic Tasks

In hierarchical LDA, a *nested Chinese Restaurant Process* (nCRP) [GG06] is used as additional prior on the topic-word distribution. Each word is assumed to be drawn from $p(\beta|\eta)$ with $\beta = (\beta_1, \dots, \beta_L)$ a partition of infinitely many distributions: $(\beta)_\infty$. The β_i , respectively the indices i , are drawn from the nCRP.

The nCRP defines a process of probability measures over infinitely many branching, (fixed or variable depth) trees. The measures put probability masses on partitions of integers. Such partitions can be interpreted as nested paths in a random tree. Starting at a random node as root node, each node is associated with a topic and a word distribution β . We identify the topics by the tuple (i, j) and the topic-word distributions by $\beta_{(i,j)}$, for a path i and a node at level j . For example in Figure 5.7 we see two paths, $c_1 = (1, 2, 3)$ and $(1, 2, 4)$, for the nodes 1, 2, 3 and 4. Each node represents a topic and the hierarchy in the tree represents the hierarchy of topics. To distinguish the path, we identify c_1 by $((1, 1) = 1, (1, 2) = 2, (1, 3) = 3)$ and $((2, 1) = 1, (2, 2) = 2, (2, 3) = 4)$. Note that the paths with common prefixes build the nesting from the nested Chinese Restaurant process.

Each path in the tree is associated with an attention curve. This means the time stamps τ of the documents for a path c_i in the random tree have distribution $SG(\tau; a_i, b_i)$. Similar to TOT, we assume that the time stamps are independent of all variables in the topic model, given a path c . In Figure 5.8 the corresponding graphical representation is shown.

The process can be summarized as the following [BGJ10]. For one document, the first token is assigned to a node with probability $p \propto n_i$, for n_i the number of tokens that have already been assigned to node i . The next token considers a random but fix child of the last node. Each node has one unique parent. Again, with probability $p \propto n_i$ the token is assigned to this child and with probability $p \propto \gamma$ to a child of this child for a smoothing parameter γ . This process repeats for all tokens in all documents and results of a random subtree of an infinite tree that is interpreted as topic hierarchic in a topic model.

The generative process of hierarchical LDA with attention curves can be summarized as

1. c_1 is common root node
2. For $i = 1, \dots, L$ and $j = 1, \dots, M$:
 - a) Draw $\beta_{(i,j)} \sim \text{Dir}(\eta)$
3. For each document $d \in D$:
 - a) Draw path c_d from nCRP
 - b) Draw $\tau_d \sim SG(\tau; c_d)$
 - c) Draw $\theta_d \sim \text{Dir}(\alpha)$
 - d) For each word w_n in document d :
 - i. Draw level $t \sim \text{Mult}(\theta_d)$
 - ii. Draw $w_n \sim \text{Mult}(\beta_{c_d(t)})$

For inference of the hierarchical LDA with attention curves, we use Gibbs sampling. We sample paths from a random tree of topics for each document. The levels encode the levels of

5. Use Case Diachronic Linguistics

the hierarchy for each token. The levels can be sampled analogue as in standard LDA the topics are sampled. For the paths, we need to define a new Gibbs sampler. The probability of a path c_d , given the words \mathbf{w}_d in document d , all other paths c_{-d} , the current topic assignments \mathbf{t} and the time stamp τ is

$$p(c_d|\mathbf{w}, c_{-d}\tau) \propto p(w_d|c, w_{-d})p(c_d|c_{-d})p(\tau|c_d).$$

The probability of a path given a sequence of words and topic assignments depends on the path, the probability of the words for the path and probability of the time stamp for the path. Given a path $c = (c_1, \dots, c_L)$, the time stamps have the distribution

$$p(\tau|c) = \prod_j p(\tau|c_j)$$

and for each time stamp we have

$$p(\tau|c_j) \sim \text{SG}(a_j, b_j).$$

This definition is analogue to the temporal topic models with attention curves, but we have attention curves associated with paths in the topic hierarchy.

For each document, a path is drawn by the nCRP. Each node in this path is associated with a β_i . For each token in the document a certain level in this path is drawn from $p(\theta|\alpha)$. Since, this is the same as in standard LDA, we can use the Gibbs sampler as before to sample the level for each word. The probability of a sequence of word w_d from document d , given paths of all other documents and the assigned levels for each token $\mathbf{c} = (c_1, \dots, c_M)$ with $c_i = (t_{i1}, \dots, t_{iL})$ is

$$\begin{aligned} p(w_d|\mathbf{c}) &= \int p(w_d, \beta|\mathbf{c})d\beta \\ &= \int p(w_d|\mathbf{c}, \beta)p(\beta|\eta)d\beta \\ &= \prod_{i,j} \frac{B(n_{i,j} + \eta)}{B(\eta)}, \end{aligned}$$

with $n_{i,j}$ the number of assigned tokens to node (i, j) , hence the node at level j in path i , $n_{i,j}^d$ the number of tokens from document d assigned to topic (i, j) and $n_{i,j}^{-d}$ the number of assigned tokens to node (i, j) without the tokens from document d , note that $n_{i,j} = n_{i,j}^{-d} + n_{i,j}^d$. Finally, the probability of a sequence of words w_d from a document d , given all other tokens from the other documents for a path c is

$$\begin{aligned} p(w_d|w_{-d}, \mathbf{c}) &= \frac{p(w_d, w_{-d}|\mathbf{c})}{p(w_{-d}|\mathbf{c})} \\ &= \prod_{i,j} \frac{B(n_{i,j} + \eta)}{B(n_{i,j}^d + \eta)}. \end{aligned}$$

5.3. Topic Models for Diachronic Linguistic Tasks

Using a log-transformation, the last equation can be simplified for the Gibbs sampler. We use that

$$\log \Gamma(n_{t,i} + \eta) = \log \prod_{j=0}^{n_{t,i}} (i + \eta) \Gamma(\eta) = \sum_{j=0}^{n_{t,i}} \log (i + \eta) + \log \Gamma(\eta)$$

and $n_{i,j} = n_{i,j}^d - n_{i,j}^{-d}$. Further we define $n_{i,j,v}$ as the number of assignments of word v to topic (i, j) , $n_{i,j,v}^d$ as the number of assignments of word v in document d to topic (i, j) , $n_{i,j,v}^{-d}$ as the number of assignments of word v of all documents but document d to topic (i, j) . The simplified probability is

$$\begin{aligned} \log p(w_d | w_{-d}, \mathbf{c}) &= \sum_{i,j} \log B(n_{i,j} + \eta) - \log B(n_{i,j}^d + \eta) \\ &= \sum_{i,j} \left(\sum_v \log \Gamma(n_{i,j,v} + \eta) - \log \Gamma\left(\sum_v (n_{i,j,v} + \eta)\right) \right. \\ &\quad \left. - \sum_v \log \Gamma(n_{i,j,v}^d + \eta) + \log \Gamma\left(\sum_v (n_{i,j,v}^d + \eta)\right) \right) \\ &= \sum_{i,j} \left(\sum_v \sum_{k=1}^{n_{i,j,v}^{-d}} \log (k + \eta) - \log \sum_v \sum_{k=1}^{n_{i,j,v}^{-d}} (k + \eta) \right). \end{aligned}$$

Using the simplify probability $\log p(w_d | w_{-d}, \mathbf{c})$, the nested Chinese Restaurant Process as defined above and the probability distribution of the attention curves we sample whole paths in a block.

During the Gibbs sampling for the paths, we generate a random tree of depth L and each node has at most M children. We distinguish two cases for the Gibbs sampler. First, we sample a path, that has been sampled before. Second, we sample a path for which only a prefix has been sampled before. We always start at the root node $(i, 1) = 1$. Hence in the second case, a random tree branches off into a new path at a certain node. Since such a new path does not have any tokens assigned to its, each such path has the same probability and we can add a new path to the random tree. This new path will be the new sample from the Gibbs sampler.

For every node (i, j) that has not been part of any sampled path so far, we have $n_{i,j,v} = 0$. This means, branching off a new path at any node at a certain level, adds the same amount to the probability of the path. This makes the sampling efficient and easy to implement. Consider, for example, a corpus with M documents and a random tree of depth $L = 3$ for the topic hierarchy estimated for the first m documents as depicted in Figure 5.7. We have two possible paths from the root to a leaf, $c_1 = (1, 2, 3)$ and $c_2 = (1, 2, 4)$; the nodes in the path c_1 and c_2 are identified by $(1, 1) = 1, (1, 2) = 2, (1, 3) = 3, (2, 1) = 1, (2, 2) = 2, (2, 4) = 4$.

Now we sample a new path for the $m + 1_{th}$ document. For simplicity assume the new document contains only one word. The probability of the path $c = (1, 2, 3)$ is

$$p_1 = \frac{n_1}{n - 1 + \gamma} \frac{n_2}{n_1 - 1 + \gamma} \frac{n_3}{n_2 - 1 + \gamma} \frac{B(n_{1,1} + \eta)}{B(1 + \eta)} \frac{B(n_{1,2} + \eta)}{B(1 + \eta)} \frac{B(n_{1,3} + \eta)}{B(1 + \eta)}.$$

5. Use Case Diachronic Linguistics

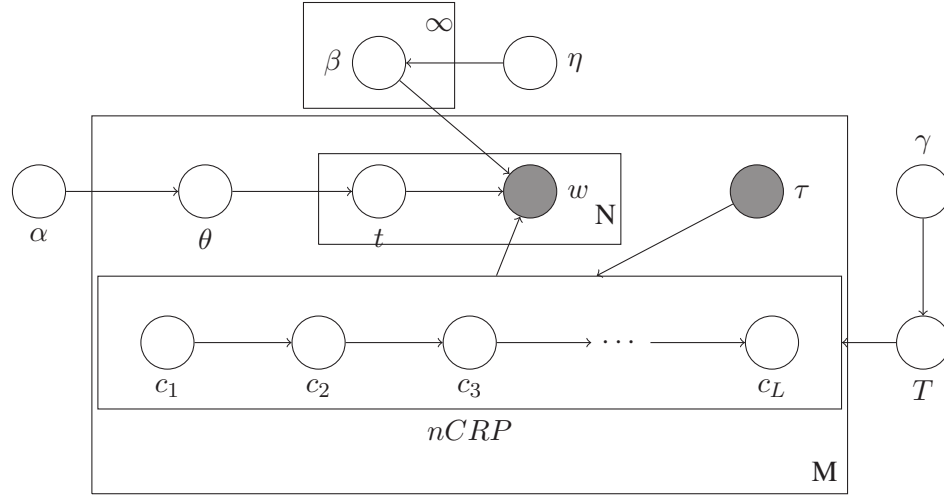


Figure 5.8.: Hierarchical Temporal LDA represented as simplified graphical model in the Plate notation.

Any path $c = (1, 2, x)$ with prefix $(1, 2)$ any $x \notin \{3, 4\}$ has probability

$$p_2 = \frac{n_1}{n-1+\gamma} \frac{n_2}{n_1-1+\gamma} \frac{\gamma}{n_2-1+\gamma} \frac{B(n_{1,1}+\eta)}{B(1+\eta)} \frac{B(n_{1,2}+\eta)}{B(1+\eta)} \frac{B(1+\eta)}{B(1+\eta)}.$$

In the sampling, we would select path c_1 with probability p_1 and a new path with prefix $(1, 2)$ with probability p_2 . In this example, a new path would be $c_3 = (1, 2, 5)$ with a new node $(3, 3) = 5$ since we get a new path c_3 (so far we had two). Analogously, the probability of a new path $(1, x, y)$ is

$$\frac{n_1}{n-1+\gamma} \frac{\gamma}{n-1+\gamma} \frac{\gamma}{n-1+\gamma} \frac{B(n_{1,1}+1+\eta)}{B(1+\eta)} \frac{1+\eta}{B(1+\eta)} \frac{B(1+\eta)}{B(1+\eta)}.$$

Sampling this path results in $c_3 = (1, 5, 6)$ with new nodes $(3, 5)$ and $(3, 6)$.

The advantage of this hierarchical model is that we can identify up and downs in the usage of certain topics over time. We can also model periodic topics that follow attention curves. Since our proposed model is non-parametric, we need no assumption of the periodicity. The number of nodes is variable and optimally estimated, only the depth is fixed.

5.4. Evaluation

We evaluate our approach to model topics over time with attention curves on several corpora and different sample sizes of documents. From the DWDS² Corpora, we use the Core-Corpus containing almost 80.000 documents with approximately 100 million tokens. Further, we use

²www.dwds.de

the Die Zeit magazine corpus containing news articles from the news magazine Die Zeit from 1947 to 2015 with more than 200 million tokens (see also the introduction in Chapter 1).

Beside the DWDS corpora, we consider further publicly available text collections. First, we use Wikipedia talk pages as provided by the Institute of the German Language³ as corpus of social media content from 2002 to 2015. Second, we use the NIPS article from 1987 to 2006, and the Union Addresses of current states of the nation from US-American presidents for quantitative comparison. Finally, we use articles from the German news magazine Spiegel from 1947 to 2013 (these are unfortunately not publicly available). For visualization, we use Andrew Goldstone's DFR-Browser for topic models <https://agoldst.github.io/dfr-browser/>. The topic numbers shown in the figures correspond to the number as given in the format used in the DFR-Browser.

From the different corpora, we retrieve documents containing content of interest with additional information about the publication date. On these document collections, we test the different temporal topic models. We compare our attention based temporal topic model, noted as @TM (for attentional topic model), with the state-of-the-art temporal topic model Topics over Time that uses a Beta distribution to model time and standard LDA that can be seen as using a uniform distribution to model time. We want to test the Shifted-Gompertz distribution whether it is better suited to extract periods of certain topics in the texts or not.

We evaluate the temporal topic models qualitatively by plotting the temporal distributions of the extracted topics, the top words within each topic and the estimated temporal distribution (Shifted-Gompertz, Beta). Quantitatively, we estimate the log-likelihood of the estimated topic models on a hold out data set. We split the data into two parts. The first part contains 80% of the whole document collection and is used to estimate the topic models. The second part contains the remaining 20% percent of the document collection and is used to estimate the log-likelihood given the temporal topic model. Additionally, we measure the coherence of the topics in terms of time and the top ranked words.

5.5. Qualitative Results

For a qualitative analysis of the temporal topic models, we investigate the extracted topics on different document collections from the corpora. First, we investigate how well different word meanings can be captured within the found topics and how these topics distribute over time. We are interested in the change of the word meanings and word usages over time as diachronic linguistic task. Second, we investigate how different subjects in the document collections change over time. Here, the diachronic linguistic task is to identify different periods of interest in the subjects.

5.5.1. Lexicography using the German Reference Corpus

In the first experiment, we investigate word senses over time as diachronic linguistic task for lexicography. From the DWDS Core-Corpus, we extract snippets of one sentence containing the German word *Platte*. Overall we have a KWIC-list of 3777 snippets from documents from

³<http://wwwl.ids-mannheim.de/kl/projekte/korpora/verfuegbarkeit.html>

5. Use Case Diachronic Linguistics

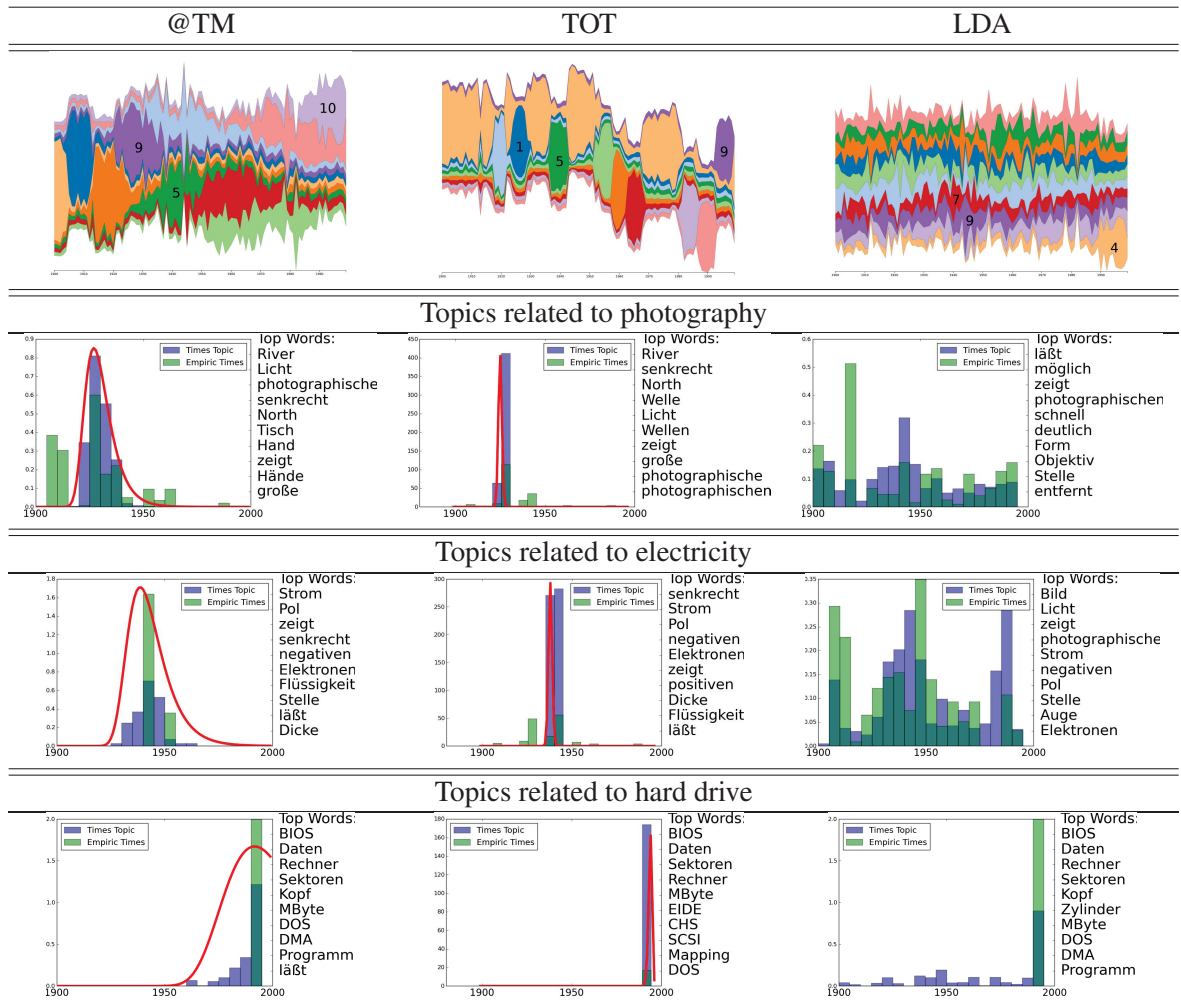


Table 5.1.: Topics found by @TM, TOT and LDA from snippets retrieved from the DWDS Core-Corpus containing the word *Platte*. Above: Topic distributions in the corpus over time (marked on the x-axis). Different colors represent different topics. Below: Distributions of time for the different topics. Each figure shows the frequency (left y-axis) of the given topic over the years (marked on the x-axis) as purple bars and the frequency of the top two words in each topic over the years as green bars. Additionally, the most important words per topic are marked on the right y-axis and the temporal distribution is plotted as red curve.

1900 to 2000. From the German dictionary, we know already possible meanings of the word: *disk*, *plate*, *music album*, *vinyl disk*. In the diachronic linguistic task, we are interested in the distribution of the different meanings of the time.

In Figure 5.1 we summarize the results of the different topic models. We show the course of the topics on the top for each model. Below, we show three hand chosen topics. For each topic, we visualize the histogram of the time stamps assigned to the topics over time by the purple bars. For TOT and @TM, we additionally plot the estimated Beta distribution, respectively Shifted-Gompertz distribution as red curve. The top 10 words for each topic are plotted on the right to each topic. Finally, we show also the histogram of the distribution of the top two words for each topic in the text collection as green bars.

From the two distributions we see that using temporal topic models we get a much clearer distinction of the topics over time. We can directly read off the topics and the temporal period when this topic was prominent. From the standard LDA, we get a much more diffuse distribution of the topics over the time.

We identify three possible main meanings in the snippets that clearly separate over time. These topics are summarized in the figures below the temporal distribution of all topics in Table 5.1. First, as shown at the bottom of the table, in topic 10 for @TM, topic 9 for TOT and topic 4 for LDA, we find computer related words as the most likely ones. The distribution of the time stamps shows a peak between 1990 and 2000. Before this period, this topic has not appeared. Topics 5 for @TM, 5 for TOT and 7 for LDA are associated with the meaning of an electronic plate that is mostly used between 1920 and 1930. Among the most likeliest words are the words *Elektronen* (Engl. electrons) and *Strom* (Engl. current). From the temporal distributions of these topics, we see that the word *Platte* in the meaning of an electronic plate is mostly present in the first half of the 20th century. In the topics 9 for @TM, 1 for TOT and 9 for LDA, the most probable words indicate the meaning of a photographic plate for the word *Platte*. Among the most likeliest words are the words *Licht* (Engl. light), *Objektiv* (Engl. objective) and *photographische* (Engl. photographic). The distribution of the time stamps shows a major usage of this meaning until the 50s.

The results from the first experiment show that the density of the Beta distribution of the time stamps tends to put too much weight on single topics. This gets worse the more topics we have since then we have less different time stamps per topic and hence the density of the corresponding Beta distribution gets very large at these time stamps. This means, the density of the Beta distribution is that large that the remaining parts of the topic probabilities are negligible. The Shifted-Gompertz on the other hand, separates the topics more smoothly and allows for several topics to exist in parallel. Comparing the extracted topics and inferred meanings from the top words, we identify also the more modern meaning of *Platte* as hard disk in a computer. By contrast, the meaning of *Platte* as music album or vinyl disk could not be found.

While the last experiment compares the different temporal topic models, we are further interested in how well our proposed attentional topic model identifies periods of high topic presence. To investigate @TM and the temporal distributions of topics related to word meanings, we perform additional experiments on the DWDS corpora. For the two words *Heimat* (home) and *Wesen* (being), we extract snippets of 3 sentences containing the corresponding words from the DWDS Core-Corpus from 1900 to 2000. For the word *Wende* (change), we extract snippets of 3

5. Use Case Diachronic Linguistics

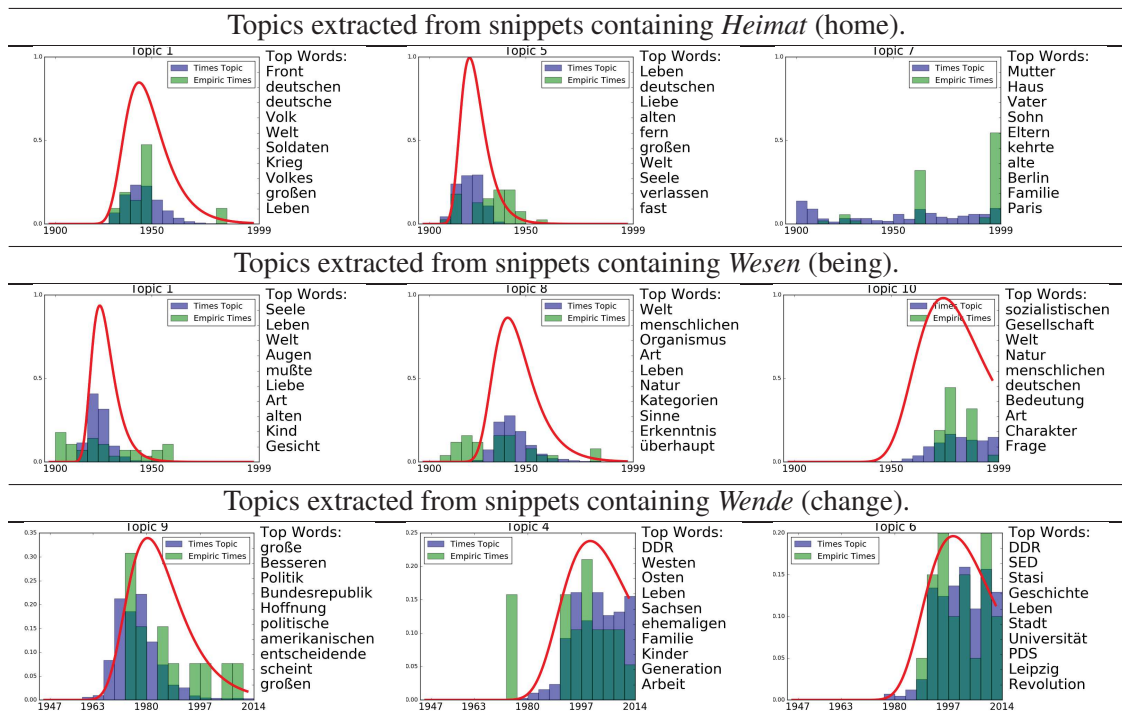


Table 5.2.: Topics found by @TM from the DWDS Core-Corpus and Die Zeit magazine corpus. Top: Topics extracted from snippets from DWDS Core-Corpus containing the word *Heimat* (home). Middle: Topics extracted from snippets from DWDS Core-Corpus containing the word *Wesen* (being). Bottom: Topics extracted from snippets from Die Zeit magazine corpus containing the word *Wende* (change). Each figure shows the frequency (left y-axis) of the given topic over the years (marked on the x-axis) as purple bars and the frequency of the top two words in each topic over the years as green bars. Additionally, the most important words per topic are marked on the right y-axis.

sentences from the Die Zeit magazine corpus from 1947 to 2015.

In Table 5.2, we report for each word three hand chosen topics from @TM. At the top: we see that the word *Heimat* was before World War II rather positively related. Between 1933 and 1970, the term *Heimat* gets connected with war. In topic 7, we find a concept of the word *Heimat* as home and family. This topic has a clear uniform distribution over the time. In the middle: the word *Wesen* is used as an expression of humans and nature especially in the first half of the 20th century. In the second half of the 20th century another concept of the word *Wesen* appears that is related with socialistic society. At the bottom: the term *Wende* is interesting since it became a metonym of the Reunification of Germany. In the 1980s, the term is used in general for changes in politics, from 1990 on, it is used primarily as reference to the Reunification of Germany.

5.5.2. Semantics in Wikipedia Discussions

In the following experiment, we investigate topics extracted from Wikipedia talk pages that contain the term *president*. The Wikipedia talk pages corpus contains comments on the articles on Wikipedia from 2002 to 2014. In Table 5.3, we show for each topic model three hand chosen topics. The topics are chosen to cover discussions about George Bush, the presidential campaign of Barack Obama and discussions about Barack Obama's heritage. In the first row, we show the distribution of topics over time (the three hand chosen topics are highlighted). Comparing these plots, we see that the uniform distribution puts equal probability on the topics at each time stamp. The Beta and the Shifted-Gompertz distribution on the other hand are able to tell topics apart over time. In the last three rows, we plot the hand chosen topics and their top 10 ranked words. We also plot two histograms of the time stamps. The purple histogram shows how many times a word with the corresponding time stamp has been assigned to the topic. The green histogram shows the number of appearances of the top two words from the topic in the whole data collection with respect to the time stamps. Additionally, we plot the densities of the Shifted-Gompertz and the Beta distribution estimated for the corresponding topic as red curve.

Comparing the uniform distribution with the Shifted-Gompertz and the Beta in the topics, we see that the uniform distribution of topic assignments reflects only the overall distribution of the words over time. The Shifted-Gompertz distribution on the other hand extracts attention periods for the topics quite accurately. The Beta distribution in turn is extremely sharp and models only topic attentions for a small peak. Here we see the disadvantage of the Beta distribution: Before and after the short peak, the topic has a probability of zero. For the topics covering discussions about Obama's presidential campaign for instance, the Beta distribution forces the topic to vanish after 2008. This is not correct since people have mentioned his campaign in the Wikipedia discussions later as well. The same is true for discussions about Obama's heritage. For the topic about Bush on the bottom of Table 5.3, we see that the Beta and the Shifted-Gompertz distribution find a topic attention period that corresponds to the term of George Bush.

5.5.3. Semantics in the Spiegel Magazine

In the next experiment, we investigate topics extracted from articles containing the word *Bundeskanzler* (chancellor) in the German magazine "Spiegel". We investigate only one of the three topics in detail and the temporal coherence for all topics containing German chancellors names

5. Use Case Diachronic Linguistics

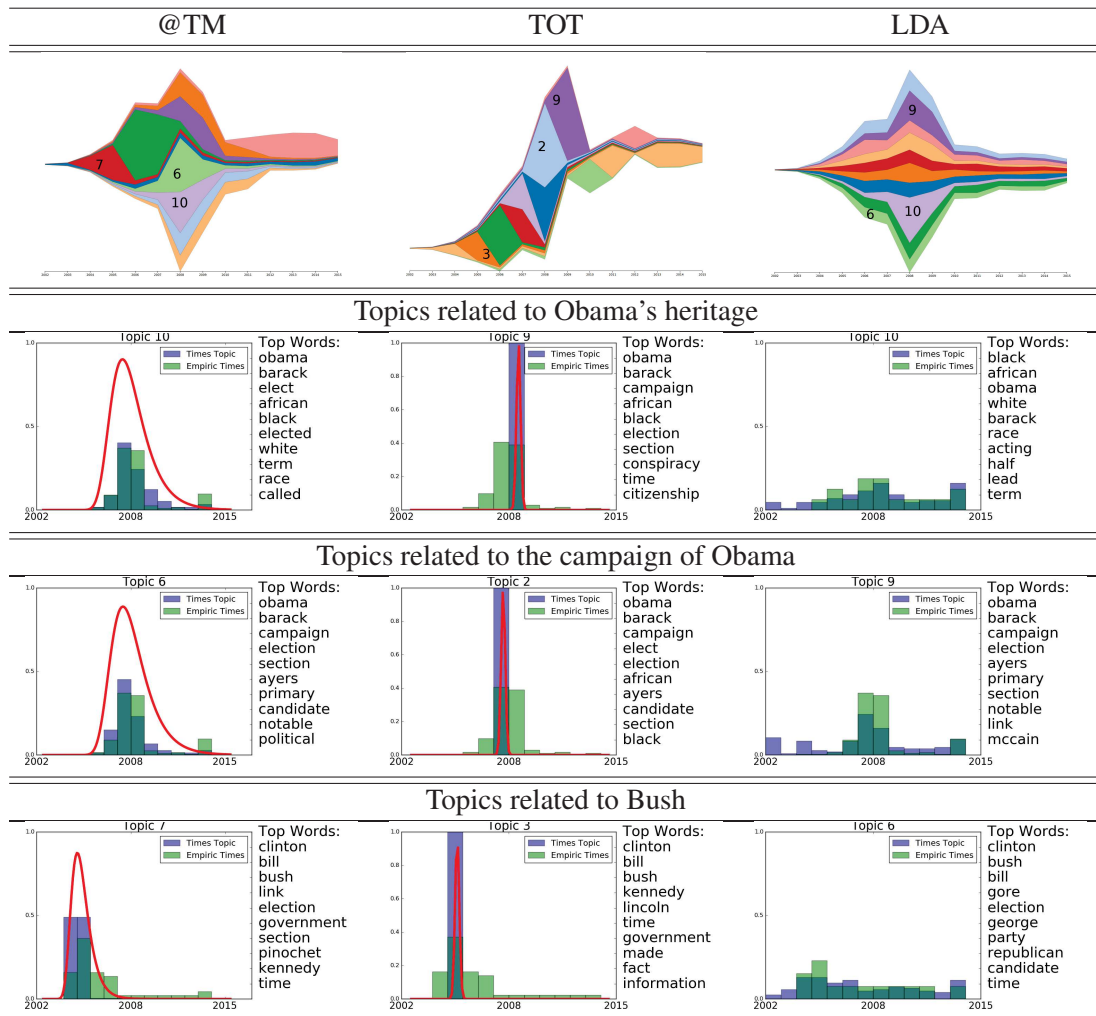


Table 5.3.: Topics found by @TM, TOT and LDA from the Wikipedia talk pages containing the word *president*. Above: Topic distributions in the corpus over time (marked on the x-axis). Different colors represent different topics. Below: Distributions of time for the different topics. Each figure shows the frequency (left y-axis) of the given topic over the years (marked on the x-axis) as purple bars and the frequency of the top two words in each topic over the years as green bars. Additionally, the most important words per topic are marked on the right y-axis.

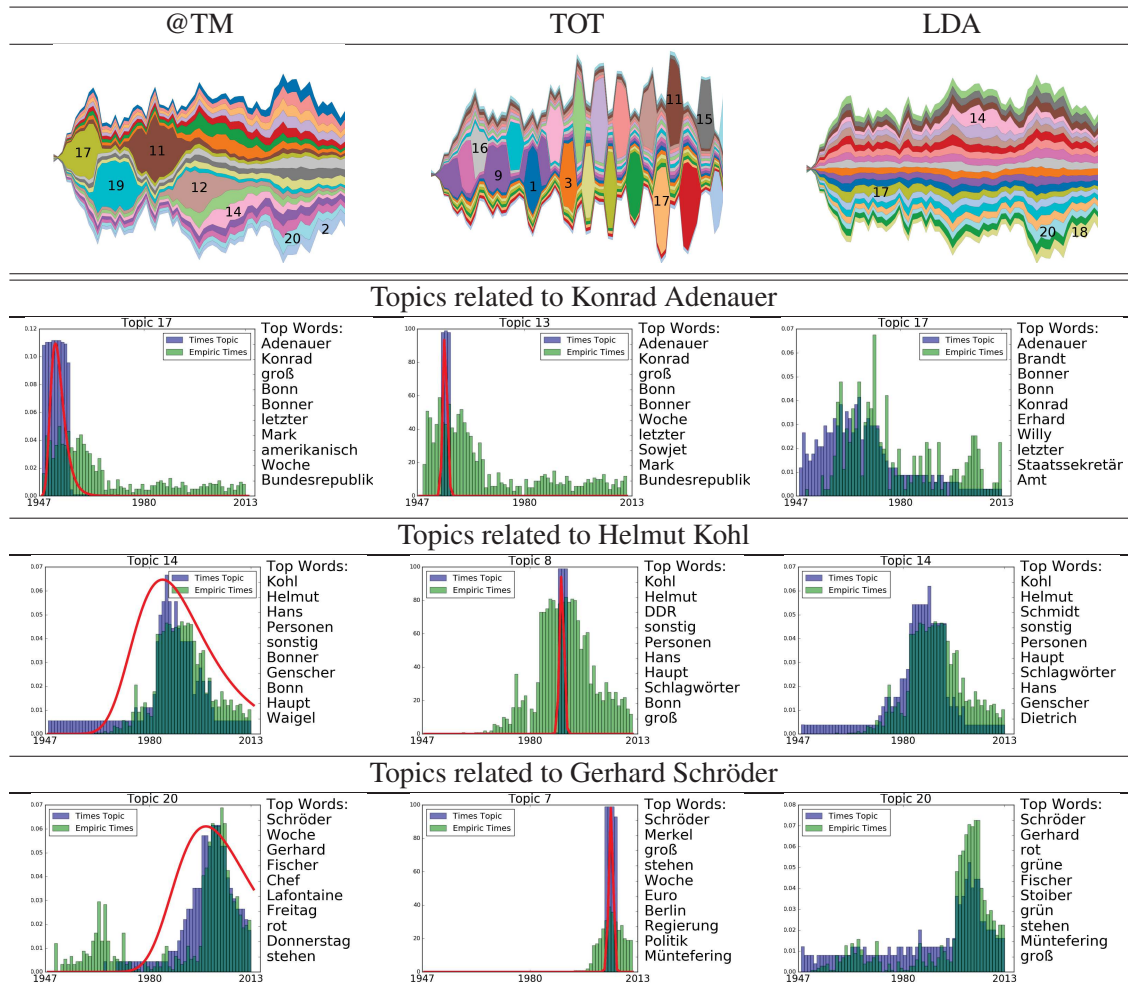


Table 5.4.: Topics found by @TM, TOT and LDA from snippets containing the word *Bundeskanzler* (chancellor) in the Spiegel corpus. Below: Topics extracted by the different topic models that are related to three German chancellors. Above: Topic distributions in the corpus over time (marked on the x-axis). Different colors represent different topics. Below: Distributions of time for the different topics. Each figure shows the frequency (left y-axis) of the given topic over the years (marked on the x-axis) as purple bars and the frequency of the top two words in each topic over the years as green bars. Additionally, the most important words per topic are marked on the right y-axis.

5. Use Case Diachronic Linguistics

	Adenauer	Erhard	Kiesinger	Brandt	Schmidt	Kohl	Schröder	Merkel
Terms	1949	1963	1966	1969	1974	1982	1998	2005
LDA	17		-	17	14		20	18
TOT	16	9	1		3	17	11	15
@TM	17	19	11		12	14	20	2
LDA	0.105		-	0.105	0.156		0.084	0.156
TOT	0.528	0.181	0.197		0.511	0.654	0.486	0.415
@TM	0.246	0.266	0.180		0.140	0.056	0.087	0.089

Table 5.5.: Temporal coherences from the topics associated with individual chancellors. Above: Terms and topic numbers assigned to each chancellor. Below: Temporal coherence measures for the corresponding topics.

among the top ranked words. In Table 5.4 we show the distribution of the topics over time in the news articles (the topics that contain any German chancellor in the top ranked words are highlighted.). Compared to the previous experiments, we get similar results in terms of the temporal distribution. The uniform distribution is not able to tell topics apart over time. The Beta and the Shifted-Gompertz distributions find topics that differ in popularity over time. Again, the Beta distribution separates the topics very sharply. Further, we see that using the Beta distribution, we get no topic that overlaps with another topic.

From 1949 to 1963, Konrad Adenauer was Germany’s first chancellor. @TM is the only temporal topic model detecting a plausible evidence of his whole term in the news articles. TOT finds a coherent topic, however it focuses all the attention on the years 1956 to 1958. LDA mixes up Adenauer’s, Erhard’s and Brandt’s terms in office. For Helmut Kohl who had his term from 1982 to 1998, @TM and LDA extract topics that cover this period. TOT on the other hand extracts only a single peak in this period at the time of the Reunification of Germany. Finally, all temporal topic models extract a topic related to Gerhard Schröder who had his term from 1998 to 2005.

We further investigate how well the different topic models are able to find periods of certain German chancellors over time, we report the German chancellors found by each topic model in Table 5.5. Using the uniform distribution, LDA finds only one big topic covering the first four chancellors but Kiesinger. TOT and @TM tell these chancellors apart. Only Kiesinger is put into the same topic as Willy Brandt. Investigating the temporal coherence values at the bottom of Table 5.5, we see that @TM is more coherent than TOT. Comparing @TM with LDA, topics over larger periods of time are more coherent when uniformly distributed. But these topics are not our main interest, as we want topics that tell periods apart. For these topics @TM results in high coherences (low values).

5.6. Quantitative Results

To quantitatively compare the different topic models, we estimate the log-likelihood of a held-out part from each of the text collections. First, we show the joint and the conditional log-likelihood for the different text collections. Additional to the Wikipedia talk pages and the Spiegel articles,

	President		Bundeskanzler	
	$\log p(w, \tau)$	$\log p(w \tau)$	$\log p(w, \tau)$	$\log p(w \tau)$
@TM	-147041	-82437	-4959928	-3281513
TOT	-144363	-87214	-4817104	-3303567
LDA	-160568	-83431	-5222189	-3256577
	NIPS		Union Addresses	
	$\log p(w, \tau)$	$\log p(w \tau)$	$\log p(w, \tau)$	$\log p(w \tau)$
@TM	-3568155	-3568155	-830967	-489864
TOT	-3581224	-3581224	-860689	-501800
LDA	-4910939	-3458160	-902309	-485450

Table 5.6.: Log-likelihoods $\log p(w, \tau)$ and conditional log-likelihoods $p(w|\tau)$ for four data sets and the different temporal topic models.

we also use the NIPS data set of papers from 1987 to 2006 and the Union Addresses data set. The last two data sets have been used in previous experiments of topic models with temporal information. For all models we set the number of topics to 10 for Wikipedia discussions about presidents and for Spiegel articles about chancellors. For the NIPS data set we use 20 topics and for the Union Addresses 40. The meta parameters from LDA are set to $\alpha = T/50$ and $\beta = 0.1$.

Table 5.7 shows the resulting likelihoods. In terms of joint likelihood, the uniform distribution has the worst results. This means, the uniform distribution is less appropriated to model time stamps together with the words in the documents. The Sifted-Gompertz results in better likelihoods for the NIPS data set and the Union Addresses, the Beta performs better for Wikipedia talk pages about presidents and for Spiegel articles about chancellors. Later, we will see that Beta has better joint likelihood when we increase the number of topics. This behavior is due to the Beta distribution that models each time stamp as a topic after a sufficient number of topics. In terms of conditional likelihood of the words given a time stamp, the Beta distribution has the lowest likelihoods. The Shifted-Gompertz and the uniform distribution result in equally high conditional likelihoods.

To investigate the likelihood in detail, we estimate the joint and the conditional likelihood for different numbers of topics. In Table 5.7 we show the course of the likelihoods for the Wikipedia talk pages and the Spiegel articles.

Finally, we compare the models by different coherence measures. As seen in Table 5.8, in terms of the standard coherence measures UMass, UCI and NPMI (cf. Section 3.2.1), we get no favored method - except for the Spiegel article data set, for which the uniform distribution results in highest coherence values. These coherence measures do not seem to be useful to evaluate topics models with time since they favor topics with top words that are constantly frequent. In contrast, the topics over time shall find topics with frequent words in certain time periods.

5.7. HLDA with Attention Curves

To evaluate the Hierarchical Topic Models with attention curves, we use a large scale corpus of conference publications. On the NIPS corpus of papers from 1987 to 2014, we apply the

5. Use Case Diachronic Linguistics

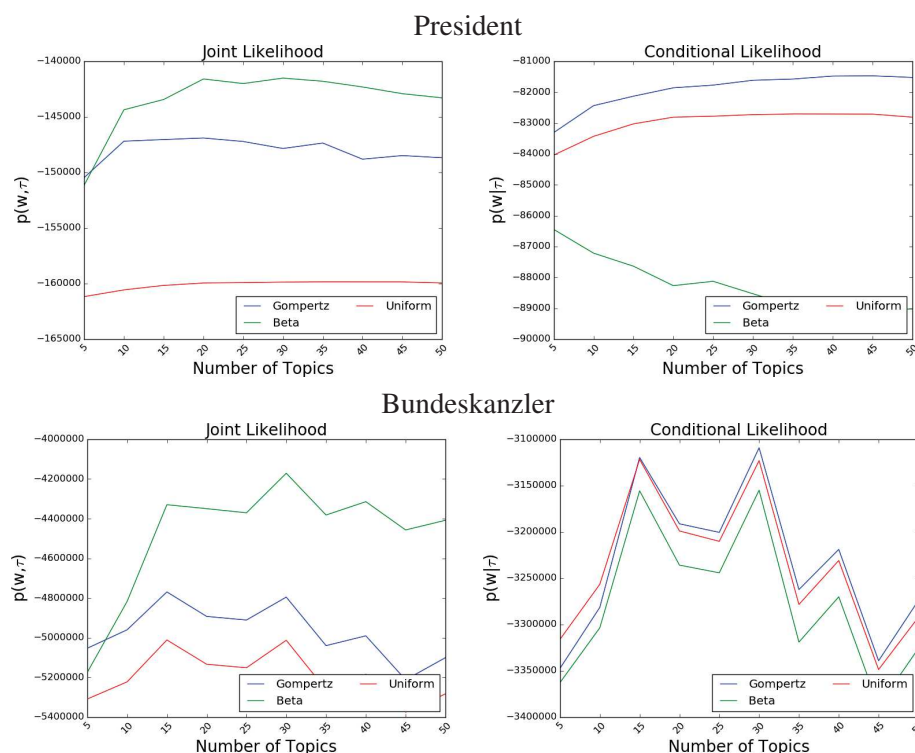


Table 5.7.: Course of the log-likelihood (y-axis) for the different topic models for varying numbers of topics (x-axis). Top: Log-likelihood of topic models for Wikipedia discussion about the *president*. Bottom: Log-likelihood of topic models for Spiegel news paper article about the *Bundeskanzler* (chancellor).

	Method	UMass	UCI	NPMI
President	@TM	-7.545	-1.545	-0.046
	TOT	-12.279	-1.551	-0.0460
	LDA	-6.353	-1.819	-0.053
Bundeskanzler	@TM	-6.241	-3.639	-0.115
	TOT	-6.492	-3.865	-0.123
	LDA	-5.836	-2.831	-0.0825
UnionAdresses	TM	-1.789	0.405	0.040
	TOT	-1.549	0.206	0.021
	LDA	-1.847	0.706	0.068
NIPS	@TM	-2.323	0.531	0.040
	TOT	-2.458	0.164	0.021
	LDA	-2.588	0.513	0.068

Table 5.8.: Standard coherence measures for topic models for four data sets. Higher scores imply better coherences of words.

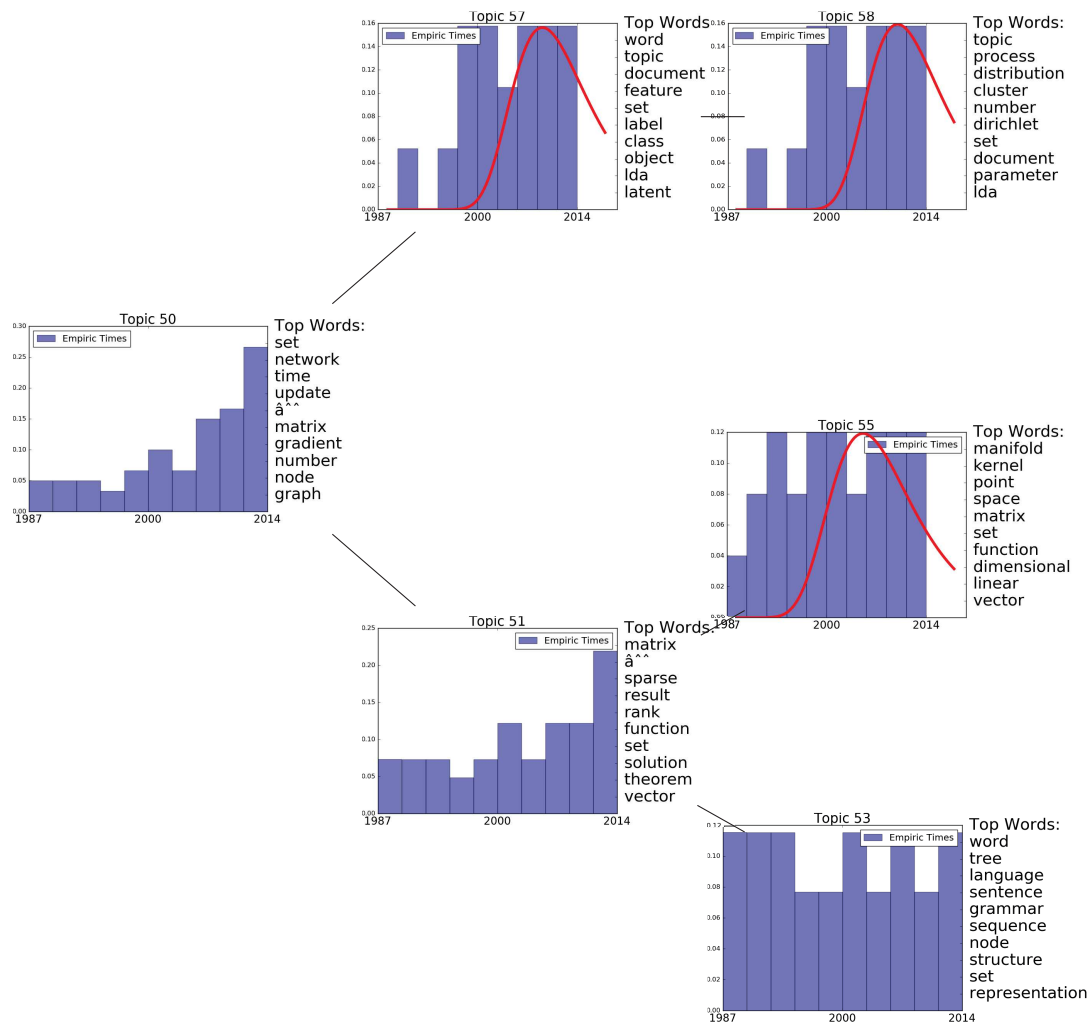


Figure 5.9.: Hierarchy of NIPS topics over time of research with respect to field of data embeddings.

hierarchical topic model with attention curves with a fixed number of $L = 4$ levels. We show the results of two hierarchies that have been extracted. As seen in the Figures 5.9 and 5.10 the hierarchy among the topics shows the refinement of general topics towards finer ones.

5.8. Conclusion

We propose a physically motivated attentional topic model. This model captures the growth and decline of topics that are popular at certain times in large text collections. For diachronic linguistic tasks in large digital corpora, we motivate and successfully apply attentional topic models. The qualitative analysis shows more informative results in terms of periods of atten-

5. Use Case Diachronic Linguistics

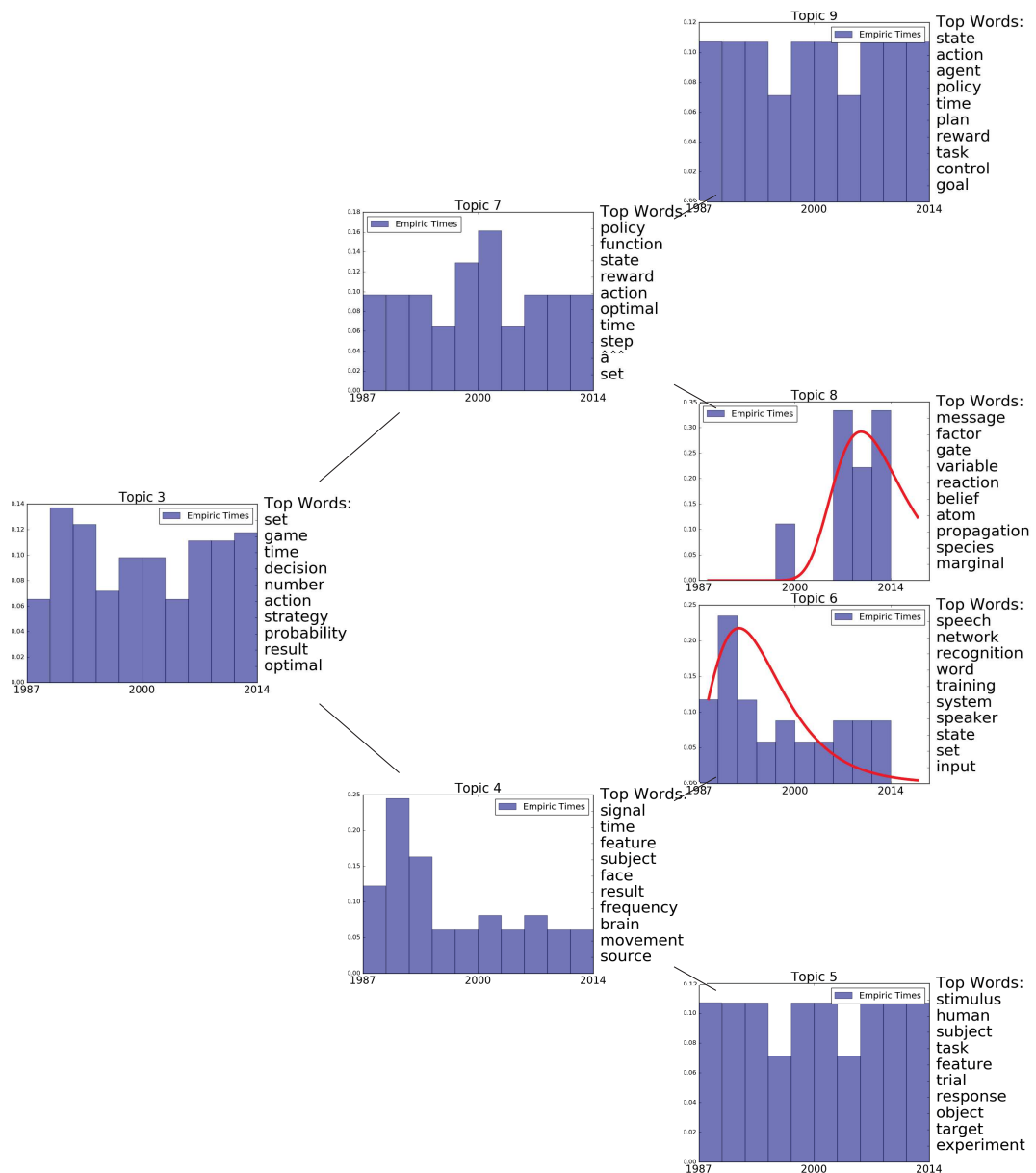


Figure 5.10.: Hierarchy of NIPS topics over time of research with respect to fields reinforcement learning.

tions to certain topics. Interpreting the topics as word senses or meanings in documents, we can model periods of affinities to certain writing styles or subjects in large digital corpora. For the quantitative analysis of topic models with time, we find standard coherence measures rather uninformative since they ignore the time information. We develop a temporal coherence measure that shows that our method finds attentional topics that coincide well with the temporal distribu-

5.8. Conclusion

tions in the corpus. Finally, from a probabilistic point of view, @TM outperforms TOT in terms of conditional likelihood. In terms of joint likelihood, @TM outperforms LDA. To estimate such likelihoods we derived a new sequential Monte Carlo Method to model the joint probabilities of words and time in documents from a corpus.



6. Use Case Non-Standard Corpora Corpus Linguistics

So far, the corpora and corresponding document collections are used as the only data source to extract latent topics, respectively factors. Certain corpora might not be complete or contain insufficient information. For the analysis of Internet-Based Communication for example, we might face document collections with scarce content. Small snippets lack enough words to effectively estimate co-occurrence statistics. The same is true if we have only a very small document collection of a view documents. Off-stream regularized latent variables can be used to include prior information into topic models such that the lack of information from the document collection gets compensated.

6.1. Motivation

The estimation of the topics highly depends on the amount of text data used. Considering the case when we have only very limited amounts of texts to estimate a topic model, the quality of the found topics can be quite poor. In such a situation, external information, or prior information, about the words can be quite beneficial. For instance, prior word probabilities can help sampling word topic distributions from a Dirichlet distribution by adding prior weights on more likely words. In this sense, we try to align the topics with an external probability model like a Language Model $p(w)$ over some of the words. External structural information like similarities of words can provide further help to align the topics. The idea is that words are similar based on external information should also be similar in the topics. This means prior weights of whole groups of similar words can be used to help estimating the topics.

To measure the quality of the found topics, intrinsic measures like the perplexity (cf. [BNJ03]) have been used in the past. Recently, coherence measures (cf. Section 3.2.1) have been introduced as an evaluation measure for topics that agree well with human judgments. These coherence measures use external information to evaluate how much related the most likeliest words in the topics are. To extract coherent topics by a topic model we must assume to have enough coherent documents. This is not always the case. In lexicography for instance, there may be rare words that appear only in a few documents. In such a case, these documents might not be enough to generate coherent topics. Further, very sparse documents as in collections of Blog posts or Tweets might also lack information to extract coherent topics.

To increase the coherence, we propose to integrate external information like word probabilities or word similarities from external data sources. To control the influence from the external information we weight this information additionally. We integrate external word probability information via Dirichlet priors similar to [MM12], but on word features instead of document features. For example, for the words of a document collection we might have the external word probabilities $p(w)$. These probabilities are integrated into a topic model such that the prior of the word distributions for each topic depends on it, hence the Dirichlet meta parameter β depends on it via the external word probabilities: $\beta \propto p(w)$. Finally, we weight these probabilities by $e^{\lambda w}$, hence we have $\beta \propto e^{\lambda w} p(w)$.

6.2. Dirichlet Priors

In the previous sections, additional label information about documents was integrated into LDA as additional random variables. Besides this, additional information about documents and words can be integrated as prior information on the random variables. In a fully Bayesian approach for LDA, the random variable θ and β have a Dirichlet prior with meta parameters. Further, the meta parameters are modeled as additional random variables. For instance, θ is Dirichlet distributed with meta parameter α , hence $\theta \sim Dir(\alpha)$. The meta parameter α itself is modeled as random variable $f(\tau)$ with distribution $p(\tau)$. Standard approaches define $f(\tau) = \alpha' a$ with $a \sim p(a)$ as basis measure and concentration parameter α' . This means, θ is a random variable drawn from a Dirichlet distribution centered at a with magnitude α' . Figure 6.1 illustrates such priors for a multinomial distribution of dimension three. We illustrate the probability simplex from

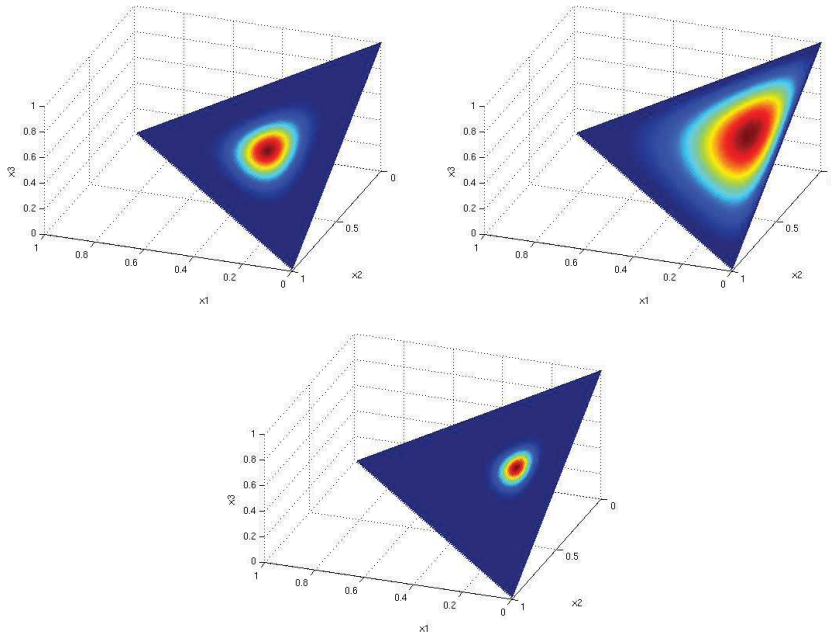


Figure 6.1.: Dirichlet priors visualized in probability simplices \mathbb{S} . The multinomial distributions from LDA lie in the set $\mathbb{S} = \{x_i | x_i \in \mathbb{R}^+, \sum_i x_i = 1\}$. The Dirichlet prior forces the multinomial distributions into certain areas marked by the colors turning to red.

which the multinomial distributions are drawn in LDA. On the top left plot, we see a probability simplex for the multinomial distribution with a symmetric prior. This corresponds to a uniform base measure. On the top right plot, we see an asymmetric prior and at the bottom we see an asymmetric prior with a larger concentration parameter. The more red like the region in the simplex is, the more likely are the corresponding distributions. See [WMM09] for a detailed analysis on symmetric and asymmetric priors for LDA.

The priors described so far are so called non-informative priors. They restrict the random variable only to sensible values independent of data. A prior is informative if the MLE differs from posterior estimation. The MLE is $p(X|\theta)$, hence the probability of the data given topic model θ . The posterior estimation is $p(\theta|X)$, hence the probability of the topic model θ given the data. Consider for instance the MLE of θ for a multinomial distribution $\text{Mult}(\theta)$, we get:

$$\theta_{dt}^* = \frac{n_{dt}}{n}. \quad (6.1)$$

Plugging this into the multinomial distribution $p(t)$, we get

$$p(t|\theta) = \theta_{dt}^* = \frac{n_{dt}}{n}. \quad (6.2)$$

6. Use Case Non-Standard Corpora Corpus Linguistics

This results from minimizing the log-likelihood given samples t_{dj} of topics (by Gibbs sampling for instance) for words w_j in documents d , with $n_{dt} = \sum_j I[t_{dj} = t]$, $n_d = \sum_t n_{dt}$ and $n = \sum_d n_d$.

Using a Dirichlet prior, the posterior on the other hand is equal to

$$p(t|\theta) = \frac{n_{dt} + \alpha_t}{n_d + \sum_{t'} \alpha_{t'}}. \quad (6.3)$$

We see that the posterior differs from the MLE such that the prior introduced pseudo counts α_t for a topic t . This encodes prior belief in the distribution of the latent topics for given documents before we have seen any word. Hence, the prior can explicitly integrate information into the topic model.

There are usually two considerations before introducing priors. First, for computational convenience priors like conjugate priors result in simpler posterior distributions. Second, insufficient data may prevent adequate estimation of the posterior and priors might compensate this. We illustrate this again with Dirichlet priors: In LDA, we assume that the topic-word distribution β has a Dirichlet prior $\text{Dir}(\eta)$. This results in the following posterior distribution for β :

$$p(\mathbf{d}|\mathbf{t}) = \int p(\mathbf{d}|\mathbf{t}, \beta) p(\beta|\eta) d\beta.$$

Due to the conjugacy this simplifies to (see Section 2.3.2)

$$p(\mathbf{d}|\mathbf{t}) = \prod_t \frac{B(n_t + \eta)}{B(\eta)}.$$

On the other hand, if we have insufficient information in the corpus to estimate the posterior, defining the right prior can help. Given for instance prior information about the word distribution $p(w)$, we can define the Dirichlet prior as $\text{Dir}(\eta_0 p(w))$. This prior information can be seen as pseudo counts for insufficient counts of the words.

Our approach assumes Dirichlet priors on θ and β and additional modeling of the meta parameters as random variables. We propose the following priors on the meta parameters of the (prior) distribution of the multinomial distributions θ and β with additional meta parameters $\mathbf{a} = [a_1, \dots, a_T] \in \mathbb{R}^T$ and $B = [\mathbf{b}_1, \dots, \mathbf{b}_T] \in \mathbb{R}^{V \times T}$:

$$\begin{aligned} p(\theta, \beta, t, \mathbf{d}|\mathbf{a}, B, \mathbf{x}_w, \mathbf{x}_d) \\ \beta_t \sim \text{Dir}(e^{\mathbf{b}_t' \mathbf{x}_w}) \\ \theta_d \sim \text{Dir}(e^{\mathbf{a}' \mathbf{x}_d}) \\ \mathbf{b}_t \sim p(\mathbf{b}_t|x_{ww'}). \end{aligned}$$

In Figure 6.2, we show the graphical representation of our proposed extension of LDA with priors that depend on external information about documents and words as features. We add document features \mathbf{x}_d , word features \mathbf{x}_w and word correlation features $x_{ww'}$ into LDA by adequate

6.3. Related Work on LDA with Additional Features

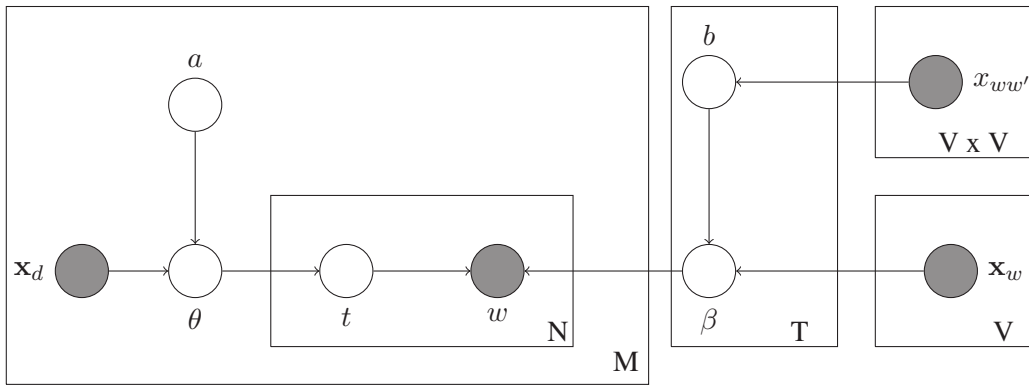


Figure 6.2.: Graphical representation of LDA with priors influenced random variables in the Plate notation. Additional document and word features are modeled as observed random variables.

priors. As before the document features \mathbf{x}_d could be time stamps or class labels, the word features \mathbf{x}_w can prior word frequency or word classes and $x_{ww'}$ are correlation information like Pointwise Mutual Information between words based on WordNet for instance.

Since we have already described how to integrate document information into factor and topic models by downstream regularized models, we concentrate here only on the integration of word information by upstream regularized models. The integration of document features into LDA can be done by DMR as described above.

6.3. Related Work on LDA with Additional Features

There are many previous approaches integrating external information into the generation of a topic model. The authors in [MM12] use a regression model on the hyper parameters of the Dirichlet prior for LDA. They use Dirichlet multinomial regression to make the prior probability of the document topic distribution dependent on document features. Analogue for the topic-word distribution, [PSC⁺10] integrate word features into LDA by adding a Logistic prior on the parameter of the Dirichlet prior of the word topic distribution. In [NBB11b] the authors integrate correlation information about words into a topic model. They propose regularized topic models that have structural priors instead of Dirichlet priors. These structural priors contain word co-occurrence statistics for instance. [MWT⁺11] propose a Pólya Urn Model to integrate co-occurrence statistics into a topic model. Finally, [AZCR11] use First Order Logic incorporated into LDA to leverage domain knowledge, [AZC09] incorporate information about words that should or should not be together in a topic from topic model, [CML⁺13] integrate lexical semantic relations like synonyms or antonyms derived from external dictionaries into a topic model.

6. Use Case Non-Standard Corpora Corpus Linguistics

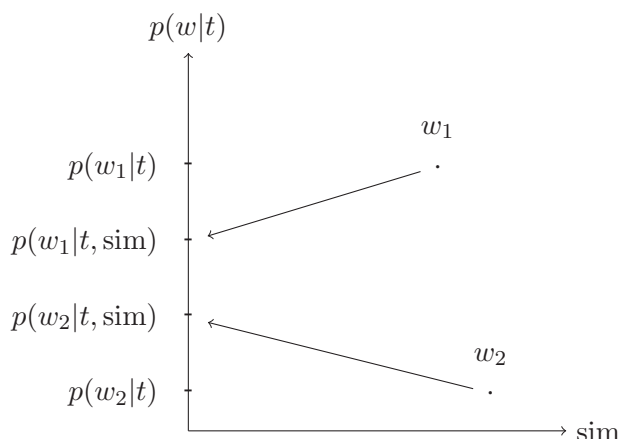


Figure 6.3.: Illustration of jointly modelling words and similarity information about the words. Words that are similar based on external information, shall have similar probabilities in the topic-word distributions.

6.4. Regularizing Topic Models by Priors

While the previous regularizations considered additional information about the documents that are corpus specific, modern language resources deliver additional corpus unspecific information about the words. In terms of lexicography and semantics, for example, this information can help explaining concepts in documents from non-standard sources with insufficient data. Documents with very few words, for instance, could lack enough co-occurrence information to extract reasonable topics. We assume to have additional information about words as word distribution $p(w)$. For latent topic models, these words shall have similar topic-word probability $p(w|t)$. Hence, similar words get similar probability mass from the multinomial distribution.

In Figure 6.3, we illustrate this principle. Words that are similar might in standard LDA have large difference in the topic-word distributions. To avoid this, the prior in the words pushes these words also closer in terms of the distributions.

We integrate external information about words into LDA via priors on the topic-word distributions. We define an asymmetric Dirichlet prior with metaparameter η on the topic-word distribution β . The parameter η specifies the prior belief on the distribution of the words before we have seen any data. A non-informative prior would set η to 1 for all words. In contrast to that, we make η informative, by making it dependent on the word distribution from the external information $p(w)$.

Formally we replace the Dirichlet prior $\beta_t \sim \text{Dir}(\eta)$ with

$$\beta_t \sim \text{Dir}(e^{\mathbf{b}'_t \mathbf{x}_w}),$$

hence $\eta = e^{\mathbf{b}'_t \mathbf{x}_w}$. The weight parameter \mathbf{b}_t controls the individual influence of the prior information in each topic for the word features \mathbf{x}_w . This parameter is a vector: $\mathbf{b}_t = (b_{t1}, \dots, b_{tV})$ where we index the weight for a word w by b_{tw} . In a fully Bayesian approach we model the weight parameter \mathbf{b}_t as additional random variable. For the specification of the distribution of

\mathbf{b}_t , we need to consider the following: If \mathbf{b}_t is zero, we get an uninformative prior. If $e^{\mathbf{b}'_t \mathbf{x}_w}$ is zero, the additional information about the words are overweighted by the likelihood of the data. On the other hand, if $e^{\mathbf{b}'_t \mathbf{x}_w}$ is very larger, the additional information overweights the likelihood of the data.

This general approach to include word features is used to add information of the expected distribution of the words. Having this distribution $p(w)$ estimated or extracted from external information, we want the topics - in terms of the β_t - to be regularized towards $p(w)$. To include prior belief in the distribution of the words, we define word features

$$\mathbf{x}_w := (\log p(w), 1)$$

and parameters

$$(1, b_{tw}).$$

This results in the following Dirichlet prior:

$$\beta_{tw} \sim \text{Dir}(e^{(1, b_{tw})' \mathbf{x}_w}) = \text{Dir}(p(w) e^{b_{tw}}).$$

This means, if b_{tw} is zero, the prior belief of the probability of w is directly used as prior for the topic-word distribution for word w and topic t . If b_{tw} is less than zero, the prior belief is weighted down. If b_{tw} is greater than zero, the prior belief is weighted up. Clearly, this model can be easily extended to integrate further features of the words.

The optimal parameters b_{tw} for each topic t must be found by optimizing the likelihood of the topic model. We perform alternating optimization of the parameters with quasi Newton methods and Gibbs sampling of topics to find the optimal topic model. For the optimization of the parameters we minimize the part of the negative log likelihood of the documents and the topics that depends on η integrating θ and β out. The negative log-likelihood that depends on η is

$$\begin{aligned} L_\eta = & \sum_t \log \Gamma(\tilde{\eta}_t + n_t) - \log \Gamma(\tilde{\eta}_t) \\ & + \sum_t \sum_{w: n_{w,t} > 0} \log \Gamma(\eta_{w,t}) - \log \Gamma(p(w) e^{b_{tw}} + n_{w,t}), \end{aligned} \quad (6.4)$$

with $\tilde{\eta}_t = \sum_w p(w) e^{b_{tw}}$.

The gradient of the negative log-likelihood is

$$\begin{aligned} \frac{\partial L}{\partial b_{tw}} = & e^{b_{tw}} p(w) (\Upsilon(\tilde{\eta}_t + n_t) \\ & - \Upsilon(\tilde{\eta}_t) + \{n_{w,t} > 0\} \cdot \Upsilon(p(w) e^{b_{tw}}) - \Upsilon(p(w) e^{b_{tw}} + n_{w,t})). \end{aligned} \quad (6.5)$$

Using these gradient informations, we can perform Newton optimization to minimize L_η . Limited Memory BFGS [LN89] for instance can be applied to minimize L_η after one Gibbs iteration.

Online Upstream Regularization

As for online LDA, we can also use off-stream regularized topic models in an online manner. Again, we separate the parameters into local and global parameters. In contrast to online LDA, we exchange the local parameter α_d to $e^{\mathbf{a}'_t \mathbf{x}_d}$ with global parameters \mathbf{a}_t and the parameter η to $e^{\mathbf{b}'_t \mathbf{x}_w}$ with global parameter \mathbf{b}_t for the topics t . As in online LDA, for each document, we estimate the local variational parameters ϕ and γ for the global parameter: $\alpha = e^{\mathbf{a}'_t \mathbf{x}_d}$ with the current estimate of \mathbf{a}_t . With these parameters we derive the global variational parameter λ with parameter $\eta = e^{\mathbf{b}'_t \mathbf{x}_w}$ and the parameters \mathbf{a}_t and \mathbf{b}_t by Maximum Likelihood Estimation.

For the global parameters \mathbf{a}_t that weight the document features \mathbf{x}_d , we have the following variational bound on the log-likelihood:

$$L = \sum_d \log \Gamma\left(\sum_t e^{\mathbf{a}'_t \mathbf{x}_d}\right) - \sum_t \log \Gamma(e^{\mathbf{a}'_t \mathbf{x}_d}) \\ + \sum_t ((e^{\mathbf{a}'_t \mathbf{x}_d} - 1)(\Psi(\gamma_{d,t}) - \Psi(\sum_{t'} \gamma_{d,t}))).$$

For the maximum likelihood estimate, we optimize this bound via gradient descent. The gradient of the global parameters \mathbf{a}_t with respect to the bound is

$$\frac{\partial L}{\partial \mathbf{a}_t} = \sum_d \mathbf{x}_d e^{\mathbf{a}'_t \mathbf{x}_d} \frac{\partial L}{\partial e^{\mathbf{a}'_t \mathbf{x}_d}},$$

with the partial derivative

$$\frac{\partial L}{\partial e^{\mathbf{a}'_t \mathbf{x}_d}} = \Psi\left(\sum_t e^{\mathbf{a}'_t \mathbf{x}_d}\right) - \Psi(e^{\mathbf{a}'_t \mathbf{x}_d}) + \Psi(\gamma_{d,t}) + \Psi\left(\sum_{t'} \gamma_{d,t}\right).$$

For the global parameters \mathbf{b}_t that weight word features \mathbf{x}_w , we get similar as for \mathbf{a}_t the following bound:

$$L = \log \Gamma\left(\sum_t e^{\mathbf{b}'_t \mathbf{x}_w}\right) - \sum_t \log \Gamma(e^{\mathbf{b}'_t \mathbf{x}_w}) \quad (6.6) \\ + \sum_t (e^{\mathbf{b}'_t \mathbf{x}_w} (\Psi(\lambda_{t,w}) - \Psi(\sum_{t'} \lambda_{t',w}))).$$

The gradient of the global parameters \mathbf{a}_t with respect to the bound is

$$\frac{\partial L}{\partial \mathbf{b}_t} = \sum_{w'} \mathbf{x}_{w'} e^{\mathbf{b}'_t \mathbf{x}_{w'}} \frac{\partial L}{\partial e^{\mathbf{b}'_t \mathbf{x}_{w'}}},$$

with the partial derivative

$$\frac{\partial L}{\partial e^{\mathbf{b}'_t \mathbf{x}_w}} = \Psi\left(\sum_{t',w'} e^{\mathbf{b}'_{t'} \mathbf{x}_{w'}}\right) - \Psi(e^{\mathbf{b}'_t \mathbf{x}_w}) + \Psi(\lambda_{t,w}) - \Psi\left(\sum_{t',w'} \lambda_{t',w'}\right).$$

Having found the optimal parameters \mathbf{a}_t and \mathbf{b}_t at iteration j , the global variational parameter λ is updated by

$$\hat{\lambda}_{tw} = e^{\mathbf{b}'_t \mathbf{x}_w} + D \sum_{n=1}^N \phi_{dn}^t w_{dn}$$

$$\lambda_{tw}^{j+1} = (1 - \rho_j) \lambda_{tw}^j + \rho_j \hat{\lambda}_{tw}.$$

Sparsity-Inducing Priors

We propose to use sparsity inducing priors on the parameters \mathbf{b} to gain control of the external information about the words. Further, we gain additional parsimony and understandability due to the sparsity. The parameter b_{tw} weights the influence of the prior information about word w for topic t . We expect that some parts of the prior information play a bigger role than other parts in the estimated topic model. To find out which parts are important we impose sparsity to identify them. This is done by adding a Laplace prior on the parameters \mathbf{b}_t :

$$p(\mathbf{b}_t; \sigma_1) = \frac{1}{2\sigma} e^{-\frac{\|\mathbf{b}_t\|_1}{\sigma}}.$$

This means, we aim to reduce the amount of off-stream regularization of the external information. This has three advantages: First, we can easily read off from the parameters which parts of the prior information influences the topics most. Second, we get a simpler model that adds the external prior information only for some words. Third, we gain control on the amount of external information to be integrated into the topic model.

Now, the process of generating documents by regularized LDA can be formulated in the following way:

1. For each topic t :
 - a) Draw $\mathbf{b}_t \sim p(\mathbf{b}_t; \sigma_1)$
 - b) Draw $\beta_t \sim \text{Dir}(e^{\mathbf{b}'_t \mathbf{x}_w})$
2. For each document d :
 - a) Draw $\theta_d \sim \text{Dir}(\alpha)$
 - b) For each word w_n in document d :
 - i. Draw $t_n \sim \text{Mult}(\theta_d)$
 - ii. Draw $w_n \sim \text{Mult}(\beta_{t_n})$

The difference to standard LDA is that we have now an asymmetric prior that is derived from the external information (for instance the word probabilities) and the weight of this information has a Laplace prior. Adding the Laplace prior on the \mathbf{b} parameters and optimizing for the negative log-likelihood is the same as putting a sparsity inducing penalty (regularizer) on them. Again, this results in the loss

$$L_\eta = \sum_t \log \frac{\Gamma(\tilde{\beta}_t)}{\Gamma(\tilde{\beta}_t + n_k)} + \sum_t \sum_{w: n_{w,t} > 0} \log \frac{\Gamma(\beta_{w,t} + n_{w,t})}{\Gamma(\beta_{w,t})}$$

6. Use Case Non-Standard Corpora Corpus Linguistics

with additional regularizations

$$R_t = \frac{\|\mathbf{b}_t\|_1}{\sigma_1}.$$

Hence in the final optimization, the negative log likelihood as defined in Equation 6.5 is extended by the term $\|\mathbf{b}_t\|_1$:

$$\arg \text{opt}_{\Theta} L_{\eta} + \sigma_1^{-1} \sum_t \|\mathbf{b}_t\|_1.$$

Hence, the Laplace prior is integrated into the optimization via a sparse lasso penalty $\|\mathbf{b}_t\|_1$. We solve the optimization problem via Orthantwise Quasi Newton Optimization [AG07] for the parameters $\Theta = [\beta, \theta, B]$.

Group-Sparsity-Inducing Priors

The previous idea of limiting the adaptation of the external prior information for some words does not consider that the information about similar words should also be treated similar. For instance, in case the prior information about the word *book* is not included for some topic, we should also not include the information about the words *author* or *books*. To formulate this idea, we divide the words into groups of similar words. The topics prior shall reflect that only word information for whole groups of words are either present or not present in the topic. The groups are noted as $g = \{w_{1,g}, \dots, w_{k,g}\}$ for groups of k words and the weight parameter is divided into parts that correspond to these words: $\mathbf{b}_g = [b_{1,g}, \dots, b_{k,g}]$.

From the group sparsity we expect more coherence since whole groups of words are considered. These groups are expected to be more coherent since they are similar based on some external information. The group sparsity prior leads to solutions with whole groups of weights either zero or are optimized to maximize the likelihood of the given texts. The groups will be specified by word similarity or co-occurrence information from different data sets.

To efficiently integrate similarity information about words, we add an additional group sparsity inducing prior on the weight vector \mathbf{b} :

$$p(\mathbf{b}; \sigma_2) = \frac{1}{2\sigma_2} e^{-\sum_g \frac{\|\mathbf{b}_g\|_2}{\sigma_2}}.$$

This prior induces sparsity of whole groups.

The resulting model adds a group lasso penalty to the negative log likelihood to gain group sparsity:

$$\arg \text{opt}_{\Theta} L_{\eta} + \sigma_1^{-1} \sum_t \|\mathbf{b}_t\|_1 + \sum_g \sigma_2^{-1} \|\mathbf{b}_g\|_2,$$

for the group lasso penalty $\sum_g \sigma_2^{-1} \|\mathbf{b}_g\|_2$ for the groups g and the variance σ_2 . Conceptionally, this is the same as having a prior on the \mathbf{b} parameters that induces group sparsity.

Similar to above we solve the group lasso via Blockwise Coordinate Descent with Proximal Operators for the group penalty, see [BJM11] for more details. After each Gibbs sampling iteration, we iterate over the groups and perform Orthantwise Quasi Newton Optimization for each group of \mathbf{b}_g keeping all other groups fix. The Newton step in the optimization is extended

with a Proximal Operator Prox to project the parameter vectors onto the group lasso constraint. Hence, after a Newton step the next \mathbf{b} is projected by the Proximal Operator

$$\text{Prox}(\mathbf{b}) = \sum_g \left(1 - \frac{\sigma_2^{-1}}{\|\mathbf{b}_g\|_2}\right)_+ \mathbf{b}_g.$$

Finding Groups

To find the groups of similar words for the grouped sparsity priors on the parameter \mathbf{m} , we use external information about similarities of words. The similarity information we use is based on WordNet (see [PPM04]). From the WordNet graph, several similarity measures can be derived. One possible similarity measure is the Leacock-Chodorow-Similarity (LCS). The LCS of two words w_i, w_j is defined as $s_{LCS}(w_i, w_j) := -\log \frac{\text{sp}(w_i, w_j)}{2D}$ with sp the shortest path between the synsets of the two words in the WordNet graph and D the maximum length of such a path, see [NLGB10].

From such similarities we can easily generate clusters that are used as groups. We divide the weight parameter $B = (\mathbf{b}_1, \dots, \mathbf{b}_G)$ into G partial weights $\mathbf{b}_g = (b_{w_1, g}, \dots, b_{w_k, g})$. The partial weights build a group g , if the words w_1, \dots, w_k build a cluster based on the similarities from the external information.

To extract the groups of similar words, we perform a clustering based on the similarity information. We generate a so called affinity matrix M such that $(M)_{ij} = e^{-(1-\text{sim}(w_i, w_j))}$ for sim the similarity measure derived from WordNet. Next, we perform a spectral clustering [NJW01] to find the groups. Spectral clustering performs a k-means clustering on the words projected onto low-dimensional space spanned by the eigenvectors of the affinity matrix. Other clustering or grouping methods are also possible but not examined in this thesis. Finally, the clusters group the words and the corresponding weights for the group sparsity prior.

6.5. Evaluation

In this section, we investigate the topics extracted by our proposed methods (SparsePrior) for LDA with sparsity prior, (GroupPrior) for LDA with group sparsity prior) and compare them with two standard state-of-the-art implementations of topic models that integrate external information about words: (RegLDA) by [NBB11b] and (WordFeatures) by [PSC⁺10]. Additionally, we also compare to the standard LDA with Gibbs sampling without external information. For each method, we use $T = 20$ topics, 1000 iterations and set $\alpha = 50/T$, $\beta = 0.1$ (for standard LDA and topic models with structural prior), $\sigma_1^{-1} = 0.1$, $\sigma_2^{-1} = 0.1$.

We use two standard text corpora used in previous approaches of topic modeling. First, we use the 20 newsgroups¹ corpus. The data set contains about 20.000 text documents from 20 different newsgroups. Overall we have 1000 documents per newsgroup. We additionally remove stop words and prune very infrequent and very frequent words. Second, we use the Senseval-3² dataset of English lexical samples. The data set contains texts from Penn Treebank II Wall

¹<http://qwone.com/~jason/20Newsgroups/>

²<http://www.senseval.org/senseval3>

6. Use Case Non-Standard Corpora Corpus Linguistics

Method	NPMI	UCI	UMASS	loglikelihood
LDA	-0.065	-2.268	-5.250	-2332131
WordFeatures	-0.061	-2.135	-4.825	-2330149
RegLDA	-0.069	-2.443	-5.520	-2332699
SparePrior	-0.070	-2.472	-5.359	-2334633
GroupPrior	-0.055	-2.116	-4.796	-2333298

Table 6.1.: Coherence results on the 20 newsgroups dataset.

Method	NPMI	UCI	UMASS	loglikelihood
LDA	0.015	-0.411	-2.534	-160480
WordFeatures	0.012	-0.465	-2.468	-160555
RegLDA	0.001	-1.767	-2.676	-160579
SparePrior	0.013	-0.579	2.8561	-160613
GroupPrior	0.020	-0.4714	-2.997	-160549

Table 6.2.: Coherence results on the Wikipedia talk pages.

Street Journal articles. The sizes of the data sets range from 20 to 200 documents per word. Further, we use the Wikipedia talk pages as social media corpus to apply our methods to a more recent data source of Internet-Based Communication. As example, we extract 10.000 postings of discussions on Wikipedia from 2002 to 2014 that contain the term "cloud".

6.6. Results

In the first experiments, we compare to the state-of-the-art LDA implementations with external information about words and standard LDA in terms of quality. We want to show that our model produces more coherent topics. To evaluate the coherence of the found topics, we use Pointwise Mutual Information (UCI), normalized Pointwise Mutual Information (NPMI) and arithmetic mean of conditional probability (UMass), see Section 3.2.1. Further, for the two larger data sets 20 newsgroups and the postings from Wikipedia we also estimate the log-likelihood on a held out data set. Finally, on the SensEval dataset, we also estimate the Mutual Information (MI) of the found topics to the true sense.

The results on the 20 newsgroups dataset in Figure 6.1 show that our proposed group sparsity prior results in topics with better coherence measures than the state-of-the-art methods and the standard LDA. From the state-of-the-art competitors only WordFeatures performs comparably good. In terms of log-likelihood, WordFeatures performs best. For the Wikipedia talk pages we get similar results as shown in Figure 6.2.

Finally, we compare the different topic model methods on collections of very small data sets. Table 6.3 shows the resulting coherence values on the SensEval dataset. LDA with our proposed grouped sparsity prior performs better on all data samples compared to the competitors.

Method	NPMI	UCI	UMASS	MI
LDA	-0.050	-1.712	-3.706	0.359
WordFeatures	-0.058	-1.744	-4.096	0.328
RegLDA	-0.056	-1.767	-3.693	0.323
SparePrior	-0.025	-0.747	-3.060	0.290
GroupPrior	-0.021	-0.634	-3.056	0.360

Table 6.3.: Coherence results on the Senseval-3 dataset.

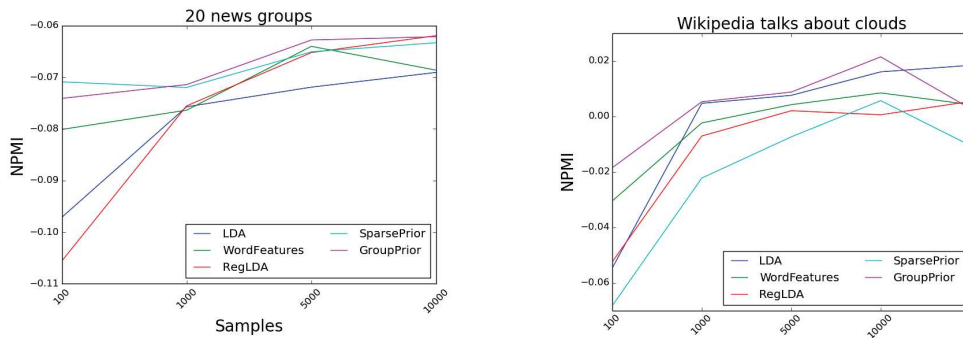


Figure 6.4.: NPMI for different sample sizes and document length used.

We are especially interested in how the different methods perform on very small data sets. To investigate this, we evaluate the NPMI for the different methods on different sample sizes and different document lengths of the samples. For the 20 newsgroups, we sample 100; 1,000; 5,000 and 10,000 documents to extract topics. From the Wikipedia talk pages we extract postings of different context sizes from 100 to 1,000 characters. In Figure 6.4, we see that our proposed sparsity and group sparsity prior result for small samples and small context sizes in the highest NPMI. In these situations our proposed method of using the group sparsity pays off the most.

6.7. Conclusion

In this use case, we propose to integrate external information about words into topic models to increase topic coherence. In non-standard corpora, we face lack of information due to sparsity in or the amount of available documents. We use different priors on the meta parameters for LDA. To control the amount of the integration of the external information, we perform an individual weighting. Adding sparsity inducing priors on these weights enables active control on how much we adapt to the external information. By this we trade off topic coherences and likelihood of the topics. Our proposed group sparsity prior further enables integration of external similarity information about words. Now, we can influence the external information of whole groups of words that are similar. The results show the benefit of our proposed methods in terms of topic coherence. Finally, we see that on very small data sets, the group sparsity inducing prior results

6. *Use Case Non-Standard Corpora Corpus Linguistics*

in better performance.



7. Use Case Variety Linguistics

For variety linguistics, we use text corpora that are composed from different text collections. Each text collection has its own writing style. They can contain documents from different genres or sources. As stated in the introduction, in variety linguistics we want to investigate different linguistic tasks in lexicography and semantics across the different text collections. In this use case, we will show how regularized factor models can be used to perform linguistic tasks across different text genre or text domains.

7.1. Motivation

For variety linguistics, information about the source or the genre of the documents in the corpus shall be integrated such that the latent variables describe the generation of documents of different sources. Especially corpora with documents from different genres can be difficult to analyze, since the contexts have different writing styles. For example, a text from an SMS differs usually

7. Use Case Variety Linguistics

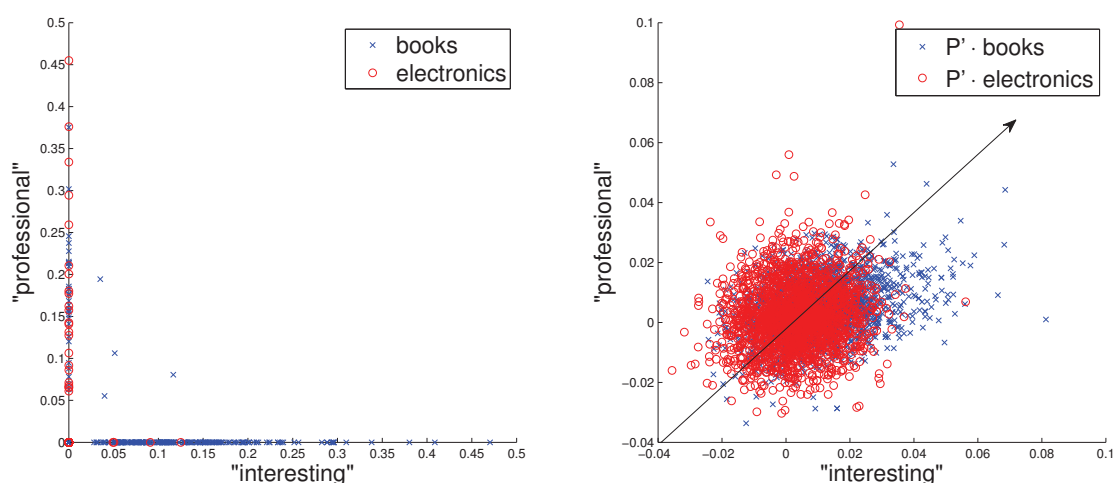


Figure 7.1.: For the VSM with word frequencies, two dimensions corresponding to the words *professional* and *interesting* are plotted. Word-vectors in the original vectors space (left) and projected space (right), from the regularization of latent factors to match document distributions from reviews about books and electronics are shown. The words *professional* and *interesting* are almost orthogonal in the original space due to the different usage of the words in the two sources. The word *professional* is used more often for electronic products, while books are described as *interesting*.

strongly from news articles. Matching the distributions from the genres helps finding latent variables that explain the generation of all documents, the commonalities and differences between document sources and genres. Given for example documents containing product reviews from two sources (hence two different products), books and electronics, we expect that words and combinations of words are differently distributed with respect to the different sources. Now, we are interested in finding concepts present in both sources or purely in one source. Jointly modeling both sources with standard latent variable models would easily end with disjunct concepts due to the distribution mismatch. This mismatch could for example lead to almost orthogonal Word-Vectors that belong to the same concept. Regularizing the factors such that the source distributions match on the subspace spanned by the factors, forces also these words into the same concepts.

In Figure 7.1, we show the distribution of the Word-Vectors for the dimensions spanned by the words *professional* and *interesting* in the VSM before and after projection into latent factors with regularization to match the distributions. On the left in the figure, we show the projection of the Word-Vectors from the reviews onto the subspace corresponding to the word *professional* and *interesting*. Although both words belong to similar positive concepts, they are differently used in reviews. Books are described rather as *interesting* while electronic article are described as *professional*. This results in almost orthogonal Word-Vectors in this subspace which makes it difficult to associate the words to the common concepts.

7.2. Projection-based Regularized Factor Models

We expect that many document collections share similarities on latent factors. For instance, a book might be described as tedious while a toaster might be described as malfunctioning. Both words have a negative connotation and very likely appear together with other negative words like *bad*, *poor* or *poorly*. Projecting the reviews onto latent factors that capture such similarities results in a subspace, on which we expect these words to jointly span a dimension representing their common ground. These latent factors represent the common concepts (e.g. sentiments) between different words from different collections, and can be expected to contain less noise. Within each document collection the factors might be different but we are interested in the common factors.

We propose to find latent factors in the space spanned by Word-Vectors that describe the similarities between document collections. This is done by a linear projection that optimally matches text documents from one collection to another collection with different document (word) distributions. The projection is performed on the Word-Vectors of the documents from the different collections and maps into a low-dimensional subspace spanned by the latent factors. Each Word-Vector from the different collections is projected onto factors \mathbf{v}^i by

$$P\mathbf{w}_d = \sum_i \omega_{id} \mathbf{v}^i = \sum_i \langle \mathbf{w}_d, \mathbf{v}^i \rangle \mathbf{v}^i.$$

In the following we no longer speak about extracting latent factors but finding projections onto the subspace spanned by the factors.

In variety linguistics, we want to match whole collections of different document collections to extract similarities and differences between the documents. Considering the example in Figure 7.1, the projection onto the subspace spanned by the factors shall make the two words more similar. As seen on the right on the figure, we want the two words to be related such that both words can be used interchangeable considering only the concept of positivity. To find these factors and the corresponding subspace, we need to consider the distribution of the words in the documents. While the word *professional* and *interesting* are almost orthogonally used in the two collections, they might appear together with additional words that bridge the collections. As seen on the top in Figure 7.2, the rather domain-neutral word *good* correlates well with the word *interesting* in book reviews and with the word *professional* in electronic products reviews. If we find projections of Word-Vectors that results in a correlation across the review collections, we expect to match even the word *professional* and *interesting* into the same concept. This is illustrated at the bottom of the figure.

In order to efficiently perform variety linguistic tasks with factor models, we propose a subspace-based regularized factor model that matches the distributions of the Word-Vectors in the factor representation. Given a number of text collections C_1, \dots, C_z , we use the regularizer

$$R(\Theta, X_d) = D(\hat{p}_1, \dots, \hat{p}_z),$$

for the document features in X_d contain information which document belongs to which collection, the empirical distributions $\hat{p}_1, \dots, \hat{p}_z$ of the Word-Vectors for each collection and the optimization parameter $\Theta = P$. A distance measure D estimates how close the word distributions of the collections are. This means, the factors are regularized such that the Word-Vectors

7. Use Case Variety Linguistics

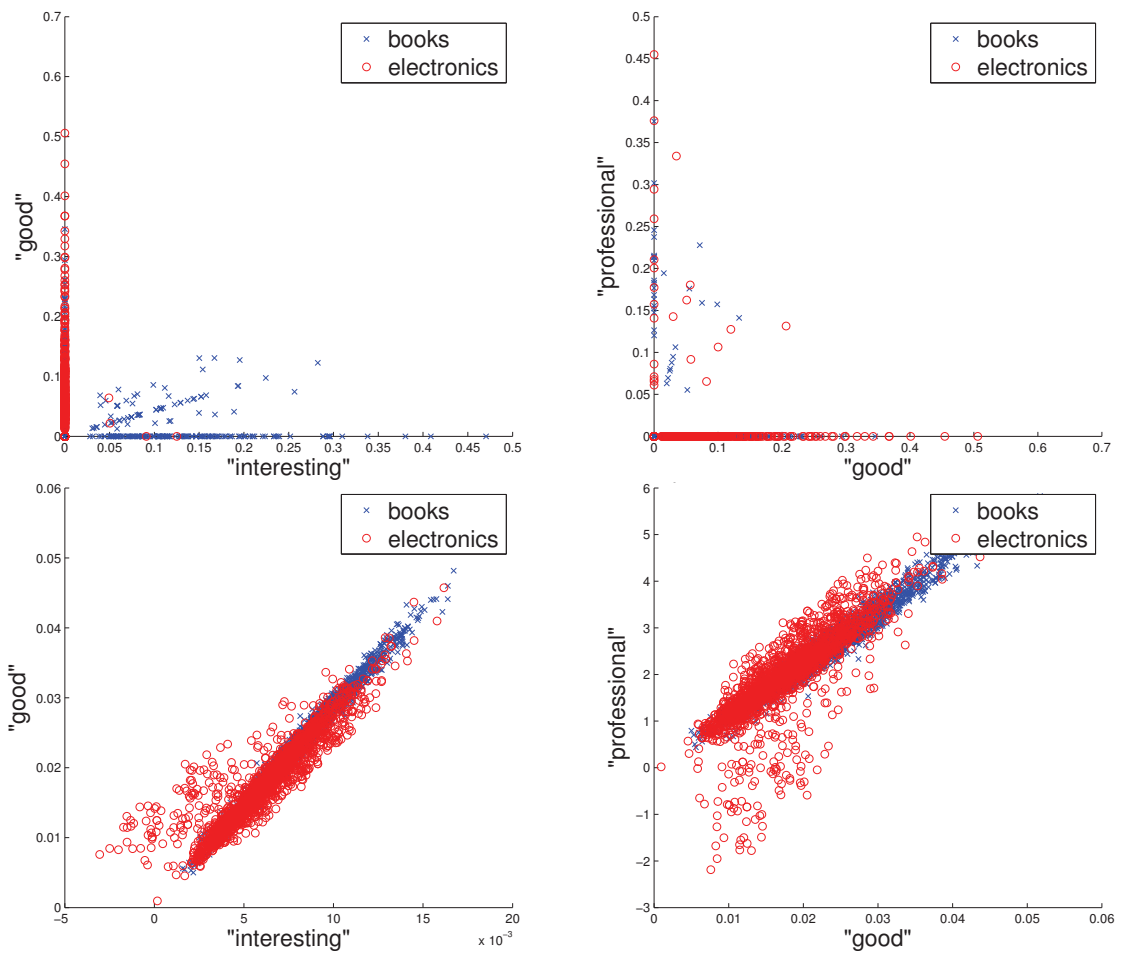


Figure 7.2.: For the VSM with word frequencies, three dimensions corresponding to the words *professional*, *interesting* and *good* are plotted. The Word-Vectors in the original feature space are plotted on the top and the Word-Vectors in projected space are plotted at the bottom. For the adaptation of “professional” to “interesting” the word “good” induces correlation.

projected via P onto the corresponding subspace have a similar distribution over all text collections. This casts the regularization of the factors into a **Domain Adaptation** task. Domain Adaptation means to make the data sets more similar to perform certain tasks. Each document collection is interpreted as a domain with its own word distribution. Matching these distributions helps finding common ground between the collections and possible dissimilarities.

7.2.1. Related Work on Domain Adaptation

Before turning to the question of transferring knowledge from one domain to another for variety linguistics, we need to discuss how to measure the distributional difference between different data sets from different domains. In the context of Domain Adaptation, divergence measures like KL-divergence [SNK⁺08b] or A-distance [BDTCP07] have been used. Such measures estimate the difference of distributions between domains of data. We use the kernelized Maximum Mean Discrepancy as proposed by [GBR⁺08] for an estimation of the difference in distribution between two data domains using samples. We do so, since this method is able to compare distributions by using all moments of the distributions. This choice is not pivotal to the contributions of this paper; it's merely a parameter that can be changed at will.

A large part of the research on Domain Adaptation concentrates on estimating weights for the target domain. Then, the data from one domain will be weighted to increase distributional similarity to another domain. Under the so-called sample selection bias, the target domain can be made similar to a source domain by an adapted weighted sampling. For instance, the authors of [DSP06] propose density estimators that incorporate sample selection bias to adapt different test domains to training domains. In [BBS09], the distance between the data from the two domains is directly minimized to find the optimal weights. The authors of [HSG⁺07a] propose to learn weights for a target domain such that the distance in distribution of the weighted target domain to a source domain is minimized. They use Kernel Mean Matching as distance measure between the domains and perform the search for optimal weights in a universal Reproducing Kernel Hilbert Space. By contrast, in [SNK⁺08a] the authors find the optimal weights via matching distributions by minimizing the KL-divergence.

Subspace-based Domain Adaptation, on the other hand, tries not to adapt distributions but to transform their support to increase similarity. This results in a low-dimensional feature representation of the original data. The transformation is done by a projection onto an appropriate subspace. In [STG10], the authors propose to minimize the Bregman divergence for regularized subspace learning. Via a matrix-variate optimization problem they find an optimal subspace for a given cost function. On this subspace, two given data sets are gauged to be similar with respect to a divergence criterion. Unlike the Stiefel approach, this optimization is directly done in \mathbb{R}^V . In [SCGF12], a low-dimensional subspace is extracted such that the data from a target domain can be expressed as linear combination of a basis from a source domain. The authors solve this problem by inexact Augmented Lagrangian Multipliers, which is computationally expensive, especially since it demands several Singular Value Decompositions (SVDs) on the data matrix. The authors of [NQC13] propose to find a sequence of subspaces in which the data from the target domain can be expressed as linear combination of a source domain. For Domain Adaptation they project all data onto each subspace and concatenate all resulting feature representations. This approach needs to perform several expensive SVDs on the data matrix. In [CLTW09] and [CSF⁺12], Domain Adaptation is coupled with the training of a classifier. The authors of [CLTW09] do this by inverting the whole data matrix, which can be quite expensive. The approach in [CSF⁺12] needs additional labels for the target domain, and a kernel matrix which might become prohibitively expensive to use.

In contrast to the linear subspaces in \mathbb{R}^V of the Word-Vectors, Kernel-based methods have also been proposed to find non-linear data representations for Domain Adaptation. In [PTKY09],

7. Use Case Variety Linguistics

Transfer Component Analysis finds low-dimensional representations in a kernel-defined Hilbert space to make two given data domains more similar. An extension of this approach by including class label information is proposed by [LWD⁺13]. The authors in [ZZW⁺13] propose to transfer knowledge in a Hilbert space by aligning a kernel with the target domain. In [MBS13], the authors propose to learn domain invariant data transformation to minimize differences in source and target domain distributions while preserving functional relations of the data.

7.2.2. Moment Matching

We concentrate on situations with two different text collections (in a given corpus) C_1 and C_2 with empirical distributions \hat{p}_1 and \hat{p}_2 and M_1 , respectively M_2 samples from the collections, for the variety linguistic task. It is straightforward to extend this to more collections. We formulate an optimization problem to find a projection matrix P onto factors together with a subspace based regularization in the following way:

$$\max_{P: P^T P = I} \lambda_1 \|PX\|_2^2 - \lambda_2 D(\hat{p}_1, \hat{p}_2, P). \quad (7.1)$$

The first part accounts for a low reconstruction error of the Word-Vectors after projection. The second part forces the empirical distributions of the Word-Vectors to be similar after projection.

In order to regularize the factors such that the distributions of the projected Word-Vectors from both collections match, we use the following regularizer:

$$D(\hat{p}_1, \hat{p}_2, P) = \left\| \frac{1}{M_1} \sum_{d_i \in C_1} P \mathbf{w}_{d_i} - \frac{1}{M_2} \sum_{d_j \in C_2} P \mathbf{w}_{d_j} \right\|_2^2.$$

This is a mean matching regularizer that punishes factors (or projections onto these factors) that result in new feature representations with large difference between the means of the collections C_1 and C_2 in this representation. We extend this to more sophisticated mean matching methods based on distances of mean operators in Hilbert space. Gretton et al. propose to use the Maximum Mean Discrepancy (MMD) [GBR⁺08] to estimate the difference in distribution between two document collections via

$$\text{MMD}[p_1, p_2]^2 = \|\mu[p_1] - \mu[p_2]\|_H^2, \quad (7.2)$$

where $\mu[p]$ is the mean operator $\int_{\mathbf{w}_d} \mathbf{w}_d dp$, p_1 and p_2 are the text distributions in the two domains and H denotes the unit ball in a universal RKHS. Hence, the MMD measures the difference of distributions as the norm in the RKHS between the means of the mappings of the distributions into this universal RKHS. In all experiments, we use Gaussian kernels, which are universal. Using a universal kernel, the MMD measures the difference based on any moment of the two distributions.

In Figure 7.3, we illustrate moment matching on a subspace for Word-Vectors belonging to documents from two different genre.

7.2. Projection-based Regularized Factor Models

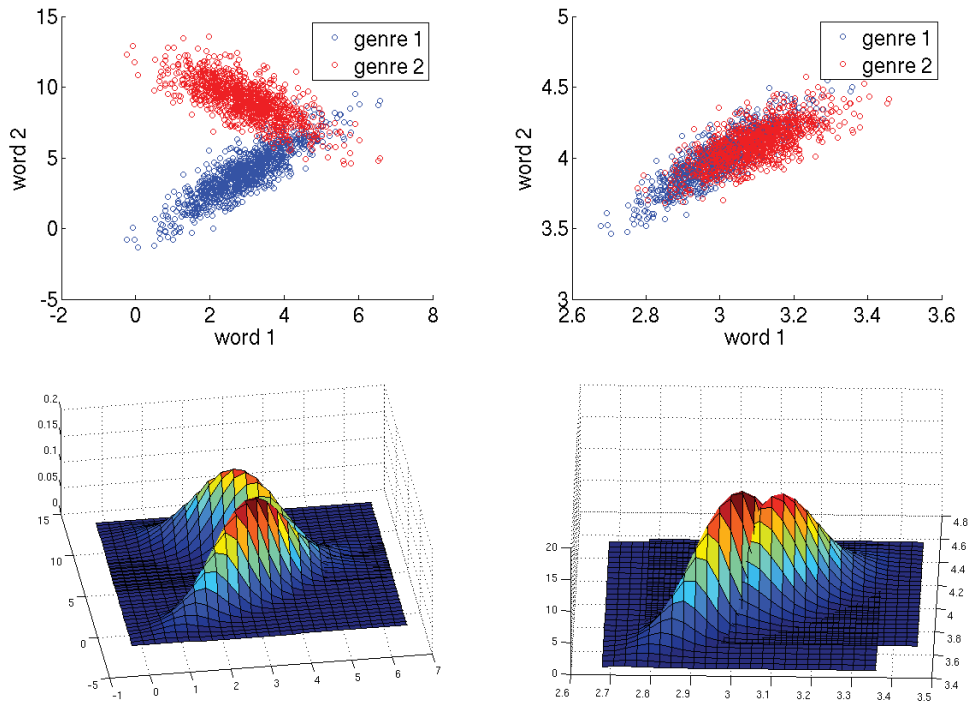


Figure 7.3.: Illustration of matching of distributions in a subspace. Top: Word-Vectors with frequency values from documents of genre 1 and genre 2 in the subspace spanned by word 1 and word 2 in the VSM. Top-left: Word-Vectors before projection. Top-right: Word-Vectors after projection for moment matching. Bottom: Visualization of the distributions of the Word-Vectors. Bottom-left: Distributions before moment matching. Bottom-right: distributions after moment matching.

7.2.3. Online Distribution Matching

We perform a Stochastic Gradient Descent (SGD) to solve the optimization problem from Equation 7.1. SGD estimates a sequence of gradients with respect to random draws from the data. Under simple conditions, this sequence converges to the optimum of the corresponding optimization problem (cf. [Bot98]). We propose an optimization that extracts interpretable linear factors based on the BoW representation of documents. At the same time we agree with the previous approaches to match the distributions of the documents based on MMD. This measure estimates the discrepancy of the two data sets based on all moments estimated from the data. This makes the problem harder, since it is no longer convex. We have no closed-form solution and must resort to gradient based approaches. The reason to apply SGD is twofold. First, we make our approach applicable to large scale scenarios. For large text collections, we resort to an online solution. Second, since our problem is non-convex and high-dimensional, we will easily end up with local optima during the optimization. SGD, in contrast to standard gradient descent (GD), adds additional randomness into the optimization that is gradually reduced in the course

7. Use Case Variety Linguistics

of the optimization. This allows to skip local minima in the beginning. The SGD is done on the Stiefel manifold $St(V, p)$ to find the optimal projection matrix that solves the optimization problem. The set $M(V, p) = \{P \mid P \in \mathfrak{R}^{V \times p}, P'P = I\}$, together with an inner product \cdot , forms a *Stiefel manifold*. A manifold is a topological space that is locally Euclidean: for each point on the manifold we find a neighborhood that is isomorphic to $\mathfrak{R}^{V \times p}$. Also, a metric is defined on each manifold that measures the distance between two points on the manifold. This local linearity and the metric enable us to define gradients to perform SGD.

Gretton et al. [GBR⁺08] describe how a linear estimation of the MMD can be defined as empirical mean over the distances of M' random draws from the two distributions in an RKHS by

$$\text{MMD}[Z]^2 = \frac{1}{M'} \sum_{j=1}^{\lfloor M'/2 \rfloor} h(z_{2j}, z_{2j+1}),$$

where $Z = \{z_1, \dots, z_{M'}\}$ is a sample of random variables $z_i = (\mathbf{w}_{d_{1i}}, \mathbf{w}_{d_{2i}})$ with the Word-Vectors $\mathbf{w}_{d_{1i}} \sim p_1$, the Word-Vectors $\mathbf{w}_{d_{2i}} \sim p_2$, and where $h(z_i, z_{i'}) = k(\mathbf{w}_{d_{1i}}, \mathbf{w}_{d_{1i'}}) - k(\mathbf{w}_{d_{1i}}, \mathbf{w}_{d_{2i'}}) - k(\mathbf{w}_{d_{1i'}}, \mathbf{w}_{d_{2i}}) + k(\mathbf{w}_{d_{2i}}, \mathbf{w}_{d_{2i'}})$ for a universal kernel $k(\cdot, \cdot)$ which induces the RKHS H . This decomposition enables us to use SGD to minimize the MMD between two distributions p_1 and p_2 .

To find the optimal projection matrix onto a low-dimensional feature representation for linear factors, we define an optimization problem that minimizes the MMD with respect to a matrix P such that $P'P = I$. The latter constraint is added to avoid rank-deficiency. Minimizing the distance with respect to a projection matrix will easily result in projections that make the data points small in length, collapse them into the origin or destroy the data structure to match the two distributions (regardless of the rank). To avoid this, we propose to regularize P via $\|PZ\|_2^2$. This leads to the optimization problem

$$\min_P \text{MMD}[Z_P]^2 - \lambda \frac{1}{M'} \sum_{j=1}^{M'} \|z'_j\|_2^2, \quad \text{s.t. } P'P = I$$

with samples $Z_P = \{z'_1, \dots, z'_M\}$ of random variables $z'_j = [P'\mathbf{w}_{d_{1j}}, P'\mathbf{w}_{d_{2j}}]$ for $\mathbf{w}_{d_{1j}} \sim p_1$ and $\mathbf{w}_{d_{2j}} \sim p_2$.

To derive a joint update rule for SGD for both the MMD and the expected length, we define the partial cost C_p of the optimization problem for the matrix $[z_{2j}, z_{2j+1}]$ drawn from Z as

$$C_p([z_{2j}, z_{2j+1}], P) := h(z'_{2j}, z'_{2j+1}) - \lambda \|[z'_{2j}, z'_{2j+1}]\|_2^2,$$

where the first term comes from the linear approximation of the MMD and the second term regularizes the length of the new feature representation for the drawn pair. The overall cost after having seen M' pairs, results from the M' partial costs

$$C(Z, P) = \frac{1}{M'} \sum_{j=1}^{M'} C_p([z_{2j}, z_{2j+1}], P).$$

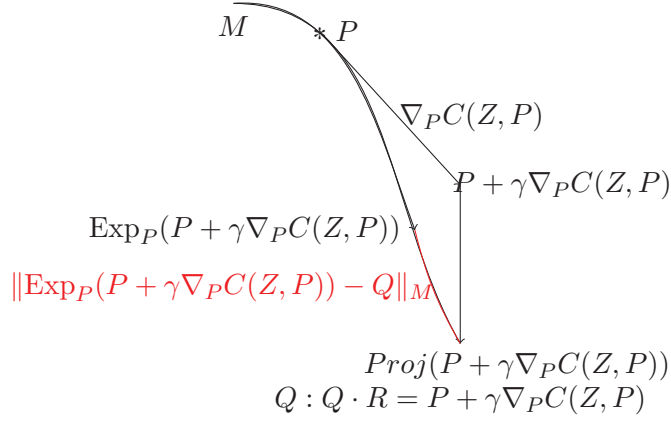


Figure 7.4.: Schematic view of an optimization step on the Stiefel manifold M . Starting at point P on the manifold, we move in the direction of the gradient $\nabla_P C(Z, P)$ (cf. Equation 7.2.3). Moving along the manifold ends in $\text{Exp}_P(P + \gamma \nabla_P C(Z, P))$. Moving simply in direction of the gradient ends in a point that must be projected back onto the manifold via (for instance) QR-decomposition. The difference of the two points is $\|\text{Exp}_P(P + \gamma \nabla_P C(Z, P)) - Q\|_M$, the norm of the difference on the Stiefel manifold.

For the SGD on the Stiefel manifold, we use the following update rule for the projection matrix P at step j [Bon13]:

$$P_{j+1} = \text{Exp}_{P_j}(H(z_j, P_j), -\gamma_j \|H(z_j, P_j)\|),$$

where H is the gradient of the cost function on the manifold. From the current projection matrix P_j , we move along the geodesic in the direction of the negative gradient of the cost function with respect to P_j . The length of the move is $\gamma_j \|H(z_j, P_j)\|$. We denote by Exp the exponential map that moves a point along the manifold in a given direction. The exponential map can be calculated in the following way [WY13]:

$$\text{Exp}_P(H, s) = \left(I + \frac{s}{2}W\right)^{-1} \left(I - \frac{s}{2}W\right) P, \quad (7.3)$$

for $W = \begin{bmatrix} H \\ P \end{bmatrix} [P, -H]$.

The major reason for directly optimizing on the Stiefel manifold is that SGD performs a large number of gradient steps. If we would not stay on the Stiefel manifold, we need to project back onto the manifold after each step due to the constraint $P'P = I$. Figure 7.4 illustrates this with a schematic view on the manifold. The curved line pictures the Stiefel manifold. At each step in the SGD we move from a current point P in the direction of a (partial) gradient $\nabla_P C$. Moving just in the direction of the gradient can result in matrices that are far away from the manifold. These matrices must be projected back onto the Stiefel manifold by a QR-decomposition for

Algorithm 8 Stochastic gradient descent on a Stiefel manifold

```

Init P
for j = 1:∞ or convergence do
  draw  $z_{2j}, z_{2j+1}$ 
  update  $P \leftarrow P + \gamma \frac{1}{j} \partial_P C_p([z_{2j}, z_{2j+1}], P)$ 
  // via exponential map or projection on the matrix manifold in direction of the negative gradient
  update  $\gamma_{j+1}$ 
end for

```

example. This results in an error $\|\text{Exp}_P(P + \gamma \nabla_P C(Z, P)) - Q\|_M$ at each step. These errors can result in slower convergence and suboptimal solutions.

Finally, for the proposed partial cost function $C_p([z_j, z_{j'}], P)$ and the next random draws $\hat{z}_j = [z_j, z_{j'}]$ from Z we get the gradient

$$\begin{aligned} H([z_j, z_{j'}], P) &= \partial_P C_p([z_j, z_{j'}], P) \\ &= \partial_P h(z_j, z_{j'}) - \lambda 2(z_j + z_{j'})'(z_j + z_{j'})P', \end{aligned}$$

consisting of the gradient of the new part of the linear approximation of the MMD and gradient of the norm of the projected data. This means that we minimize the distance on any two samples from the target and the source domain in Z , which are projected into a low-dimensional subspace, in a universal RKHS, while maximizing their length.

The gradient of h depends on the used kernel. For the Gaussian kernel k on the projected points, for instance, this results in the following kernel definition with respect to the projection matrix P :

$$k(P' \mathbf{w}_{d_1}, P' \mathbf{w}_{d_2}) = \exp\left(-\frac{(\mathbf{w}_{d_1} - \mathbf{w}_{d_2})' P P' (\mathbf{w}_{d_1} - \mathbf{w}_{d_2})}{2\sigma^2}\right).$$

The gradient of this kernel with respect to the projection matrix P is

$$\begin{aligned} \partial_P k(P' \mathbf{w}_{d_1}, P' \mathbf{w}_{d_2}) &= \\ -\frac{1}{\sigma^2} k(P' \mathbf{w}_{d_1}, P' \mathbf{w}_{d_2}) &(\mathbf{w}_{d_1} - \mathbf{w}_{d_2})' (\mathbf{w}_{d_1} - \mathbf{w}_{d_2}) P'. \end{aligned}$$

The whole optimization procedure is summarized in Algorithm 8. Here, we use a similar approach as proposed by [QPS09].

7.2.4. Related Manifold Methods

We use optimization directly on matrix manifolds. A general introduction can be found in [AMS08]. An early work on such optimization is [EAS99]. The authors develop a gradient-based optimization method on Grassmann and Stiefel manifolds. They provide a general framework for the optimization on these matrix manifolds. Both [BNR10] and [Bon13] describe a stochastic gradient descent on Riemann manifolds and illustrate its use for subspace tracking and optimization on matrices with rank constraints.

The authors of [Gra12] and [GGS13] perform Domain Adaptation on manifolds. They project the data onto all subspaces that lie on the shortest path (geodesic) between two subspaces from, respectively, the source and target domain. They define a kernel on the concatenation of all projections to extract a new feature representation. In [GLC11], the authors sample interpolated subspaces on the Grassmann manifold between a target and a source subspace, extracting domain, intermediate, and possible invariant information. Projections onto subspace samples transform the data into new feature representations. They sample these subspaces, and use projections onto these samples to transform the data into new feature representations. The authors of [BHLS13] perform gradient descent on a Grassmann manifold to find a subspace where the two given data domains have a low distance.

7.3. Regularized Non-linear Factor Models for Variety Linguistics

So far we concentrated on linear factor models for the linguistic tasks. Non-linear factor models on the other hand can be used to extract factorizations of the document in representations beyond the Bag-of-Words (cf. Section 2.2.4). Using Polynomial kernels for instance, we can map a document from its BoW representation into a new feature representation that contains collocation information of the words. For example a document containing the words “*this, is, true*” is represented as BoW $(\dots, tf_{is}, \dots, tf_{this}, \dots, tf_{true}, \dots)$ with term frequencies tf for each word. If we apply a polynomial kernel of degree two we get a new document representation $(\dots, tf_{is}, \dots, tf_{this}, \dots, tf_{true}, \dots, tf_{is}tf_{this}, \dots, tf_{this}tf_{is}, \dots, tf_{this}tf_{true})$ with collocation information. The factors are now linear combinations of these new feature representation of the documents. As described above, we can use Kernel Principal Component Analysis to extract non-linear components.

While kernel methods seem to be powerful enough to compensate the weaknesses of linear factor models, their computational and space complexity can be prohibitively expensive. There are two approaches to reduce the amount of computation and storage for kernel based methods. First, we can reduce the amount of documents used from the text corpus to reduce complexity. We can actively sample documents, that are most promising for solving the linguistic tasks. Second, we can approximate the features maps induced by the kernels by linear (random) features.

Next, we discuss how kernels can be approximated by random features and we explain document sampling to match distribution to regularize non-linear factors for variety linguistics.

7.3.1. Approximating Kernels via Random Features

To avoid large computational and storage complexity of kernel methods, approximations of the kernel can be used. Random features for instance approximate the feature maps in Hilbert spaces by low dimensional random projections. The expectation of the inner products of these random features evaluate to corresponding kernel values. Any shift-invariant kernel (as for example the Gaussian kernel) can be represented as expectation of random features $\cos(\omega'w_d + b)$ for an appropriate distribution $p(\omega)$ and b uniformly drawn from $[0, 2\pi]$, see [RR08]. For Gaussian kernels, ω is drawn from the distribution: $p(\omega) =$

7. Use Case Variety Linguistics

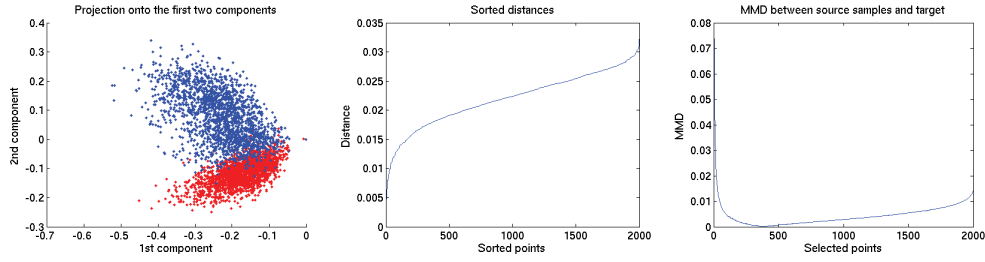


Figure 7.5.: Illustration of the samplings. Left: Source (electronic reviews in red) and target (DVD reviews in blue) data plotted in the space of the first two components of both of them together. Middle: Distances of source points to target mean (sorted). Right: MMD of the selected samples from the source data by Herding based sampling.

$(2\pi)^{-V/2}e^{-\|\omega\|^2/2}$. An unbiased estimate of the kernel is $z_\omega(\mathbf{w}_{d_i})'z_\omega(\mathbf{w}_{d_j})$ for $z_\omega(\mathbf{w}_d) = \frac{\sqrt{2}}{\kappa}[\cos(\omega'_1 \mathbf{w}_d), \dots, \cos(\omega'_\kappa \mathbf{w}_d), \sin(\omega'_1 \mathbf{w}_d), \dots, \sin(\omega'_\kappa \mathbf{w}_d)]$.

The deviation of the inner product of the random features of dimension κ to the true kernel value is bounded by a tail bound using Hoeffding's inequality [Hoe63]. Since $z_\omega \in [-\sqrt{2}, \sqrt{2}]$, we have $z_\omega(\mathbf{w}_{d_i})'z_\omega(\mathbf{w}_{d_j}) \in [-2, 2]$. This and $E_\omega[z_\omega(\mathbf{w}_{d_i})'z_\omega(\mathbf{w}_{d_j})] = k(\mathbf{w}_{d_i}, \mathbf{w}_{d_j})$ justifies the bound

$$P(|z_\omega(\mathbf{w}_{d_i})'z_\omega(\mathbf{w}_{d_j}) - k(\mathbf{w}_{d_i}, \mathbf{w}_{d_j})| \geq \epsilon) \leq 2e^{-\kappa\epsilon^2/8}.$$

The difference between the true kernel and the approximated kernel is important to quantify the expected error we will get due to approximation instead of using the kernel directly. We need to estimate how much the components for the random features deviate from the true components the source samples in the RKHS. For this it suffices to investigate the expected difference of the true kernel matrix K for M data points and the matrix of the inner products of the random features K_ω . An appropriate bound is proposed by [LPSS⁺14]:

$$E(\|K_\omega - K\|) \leq \sqrt{\frac{2M^2 \log M}{\kappa}} + \sqrt{\frac{2M \log M}{\kappa}}.$$

7.3.2. Non-linear Factors by Distribution Matching

For variety linguistics interpreted as Domain Adaptation as described above, we use subspace based regularized non-linear factor models to find non-linear factors across document collections. We do not directly regularize the factors but influence the extraction by choosing the right document samples. By this we regularize the subspace spanned by the non-linear factors to the subspace in which only these samples lie. This is important since we can not directly evaluate the factors since they are infinite dimensional. For a variety linguistic task, we project all data onto a low dimensional subspace that captures the structure of the document collections.

7.3. Regularized Non-linear Factor Models for Variety Linguistics

To find the most promising documents from a given single document collection for the Domain Adaptation, we propose a greedy strategy to efficiently select them. The sampled documents shall be close to the other document collections. On the other hand, the samples must keep enough structure of the given document collections to extract meaningful factors. The proposed strategy is based on the distance of the the source domain distribution to the target domain distribution. The picture in Figure 7.5 illustrates our idea on electronic (red) and DVD (blue) reviews. We assume the reviews of electronics as target domain and the reviews about DVDs as source domain. The reviews seem to be more similar on one direction than on the other. The idea now is, to prefer points from the source domain that are more prominent in this direction for the Domain Adaptation.

We propose a sampling strategy that is based on the data distribution. In a Hilbert space we iteratively select mapped samples from a given document collection that are most similar to the another document collection. For μ_{p_2} the expectation functional for the document collection C_2 mapped in an RKHS, the difference

$$\|\mu_{p_2} - \frac{1}{M'} \sum_{\mathbf{w}_d \in S' \subset C_1} \phi(\mathbf{w}_d)\|_H^2$$

estimates the difference of a subset of M' samples from document collection C_1 and another document collection C_2 . Similar approaches are proposed by [CWS12]. The authors showed that the sampling strategy introduced by [Wel09] can be used to match empirical and true distributions in an RKHS. This is analogue to our approach of matching distributions to find linear factors. The difference lies in how we extract the factors, respectively the subspace spanned by the factors. Here, we consider the dual case and describe the subspace by the document mappings into an RKHS. Starting with an appropriate ω_0 , we select points by the following methodology:

$$\begin{aligned} \mathbf{w}_{d_j} &= \arg \max_{\mathbf{w}_d \in C_1 \setminus S'} \langle \omega_j, \phi(\mathbf{w}_d) \rangle \\ S' &= \{\mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_j}\} \\ \omega_{j+1} &= \omega_t + E_{p_2}[\phi(\mathbf{w}_d)] - \phi(\mathbf{w}_{d_j}). \end{aligned} \tag{7.4}$$

For deciding when to stop the sampling, we monitor $\max_{\mathbf{w}_d \in C_1 \setminus S'} \langle \omega_j, \phi(\mathbf{w}_d) \rangle$. As soon as we have only data points from the source data set left such that the distance in distribution no longer decreasing, we stop. Hence, we sample points such that the empirical distributions of samples and the target data are minimal. The picture on the right of Figure 7.5 shows an example of the course of the MMD of the samples from the source domain (electronic reviews) and the target domain (DVD reviews). We sample as long as the MMD decreases to find all points that make the distribution similar. This beware us to sample points that make the two distribution dissimilar.

For $\mu_{p_2} = \frac{1}{M_t} \sum_{\mathbf{w}_{d_i} \in C_2} \phi(\mathbf{w}_{d_i})$, our sampling strategy minimizes

$$E = \|\mu_{p_2} - \frac{1}{M'} \sum_{\mathbf{w}_{d_j} \in S'} \phi(\mathbf{w}_{d_j})\|_H^2.$$

7. Use Case Variety Linguistics

To see this, we rewrite

$$E = \langle \mu_{p_2}, \mu_{p_2} \rangle - \frac{2}{M'} \sum_{\mathbf{w}_{d_j} \in S'} \langle \mu_{p_t}, \phi(\mathbf{w}_{d_j}) \rangle + \frac{1}{M'^2} \sum_{\mathbf{w}_{d_i}, \mathbf{w}_{d_j} \in S'} \langle \phi(\mathbf{w}_{d_i}), \phi(\mathbf{w}_{d_j}) \rangle,$$

with $S' = \{\mathbf{w}_{d_1}, \dots, \mathbf{w}_{d_{M'}}\}$. Since $\langle \mu_{p_t}, \mu_{p_t} \rangle$ is constant, minimizing E is the same as maximizing $\frac{2}{M'} \sum_{\mathbf{w}_{d_j} \in S'} \langle \mu_{p_t}, \phi(\mathbf{w}_{d_j}) \rangle - \frac{1}{M'^2} \sum_{\mathbf{w}_{d_i}, \mathbf{w}_{d_j} \in S'} \langle \phi(\mathbf{w}_{d_i}), \phi(\mathbf{w}_{d_j}) \rangle$. Multiplying the last expression by M' results in the greedy sampling as defined above when we set $\omega_0 = \mu_{p_2}$. This means the strategy matches the empirical distribution of the target samples with the empirical distribution of the subset of the samples from the source distribution.

Our proposed sampling strategy can still result in a large number of points from the source distribution. We further propose to combine the selection strategy and the Domain Adaptation on a subspace by random features of dimension κ . This enables us to perform the Domain Adaptation task in the linear space spanned by the random Fourier bases of the random features as defined above.

We define MMD_ω similar to MMD in Equation 7.2 except that the kernel evaluations are replaced by the inner products of the random features. Since $\text{MMD}_\omega \in [-8, 8]$, we can apply Hoeffding's inequality to bound the difference to the true MMD by

$$P(|\text{MMD}_\omega^2 - \text{MMD}^2| \leq \epsilon) \leq 2e^{-\kappa\epsilon^2/128}.$$

Due to linearity of the expectation we have $E_\omega \text{MMD}_\omega^2 = \text{MMD}^2$ and from the definition of the random features we have $k(\mathbf{w}_{d_i}, \mathbf{w}_{d_j}) = E_\omega [z_\omega(\mathbf{w}_{d_i})' z_\omega(\mathbf{w}_{d_j})]$. All together results in the bound.

7.4. Evaluation

We test the proposed method to find projection matrices by projection based regularized factor models on several corpora with different document collections. First, we use the *Die Zeit Magazine* corpus and the Wikipedia corpus with the discussion pages to compare two text collections. We apply a projection base regularization to find a subspace that is suited to explain both text collections in the vector space spanned by the Word-Vectors. We call this subspace a latent subspace since it is spanned by latent factors. Further, we use the *Amazon* review corpus containing reviews [BDP07] about products from the categories books (B), DVDs (D), electronics (E) and kitchen (K). For these corpora, we evaluate the results of the regularization qualitatively by investigating the vectors space and the Word-Vectors.

Further, we use the *Amazon* reviews to perform a quantitative analysis of the regularization. To quantitatively estimate the quality of the regularization, we perform a classification of the reviews with respect to its sentiment given in the dataset. Additionally, we use *Reuters-21578* data [LYRL04]. It contains texts about categories like organizations, people and places. For each two of these categories a classification task is set up to distinguish texts by category. Each category is further split into subcategories and different subcategories are used as source and

target domains. We denote the categories Organization by C1, Places by C2 and People by C3. The third dataset is *20 newsgroups text dataset*¹. We use the top-four categories (comp, rec, sci, and talk) in the experiments. Again, we set up a classification task for each pair of categories. Each category is further split into subcategories and different subcategories are used as source and target domains; each such configuration is denoted by Conf i . Documents of categories comp and rec shall be distinguished in Conf1, comp and sci in Conf2, comp and talk in Conf3, rec and sci in Conf4, rec and talk in Conf5, and sci and talk in Conf6.

7.5. Qualitative Results

In order to qualitatively analysis the regularization by moment matching on a subspace, we inspect projections in the vector space of the Word-Vectors. An advantage of using linear projections to find low-dimensional latent feature representations for Domain Adaptation is that they are interpretable. The projection is performed in the vector space that is spanned by the words. Hence, the projection in the individual dimensions corresponds to the word adaptation required to make two domains similar in distribution. The Word-Vectors are rotated and stretched, where the stretching is limited due to the regularization on the feature vector sizes. The amount of rotation in the vector space in certain dimensions tells how much individual words need to be adapted (of weighted). We can gauge how strongly individual words need to be adapted by inspecting the magnitude of the rotation in the vector space in the corresponding dimension.

Figure 7.6 illustrates this concept with an artificial example. In two dimensions of a vector space, Word-Vectors of two domains from different document collections are plotted. Each axis displays the normalized term frequency values in one component; each component tells the frequency of a certain word in a document multiplied by a normalization term. The SGD on the Stiefel manifold method finds latent subspaces such as the diagonal line in the figure. Projecting the vectors from both domains onto this space via the found projection matrix P implies rotating the Word-Vectors. The vectors for *word 1* and *word 2* are rotated to bridge domains. The average rotation required for the red circles is lower than the average rotation required for the blue circles. Hence, although both words are important to bridge domains, *word 2* is more different in distribution in the two domains than *word 1*. If we find little or no rotation in some dimensions, we conclude that the corresponding words are less different distributed. In the experimental section, we explore this concept on concrete real-world results.

7.5.1. Lexicographic Varieties across Social Media and Press Media

In the first experiment, we compare two document collections with different writing styles. From the social media corpus of Wikipedia talk pages, we retrieve postings containing the word *Krise* (crisis). Similar, we retrieve snippets from articles from the Die Zeit magazine containing the word *Krise*. We want to investigate the similarities and dissimilarities in the usage of this word in the document collections. Initially, we extract 3 topics from each document collection by the attentional topic model to get an overview. In Table 7.1, we show the topics extracted from the collections.

¹<http://qwone.com/~jason/20Newsgroups/>

7. Use Case Variety Linguistics

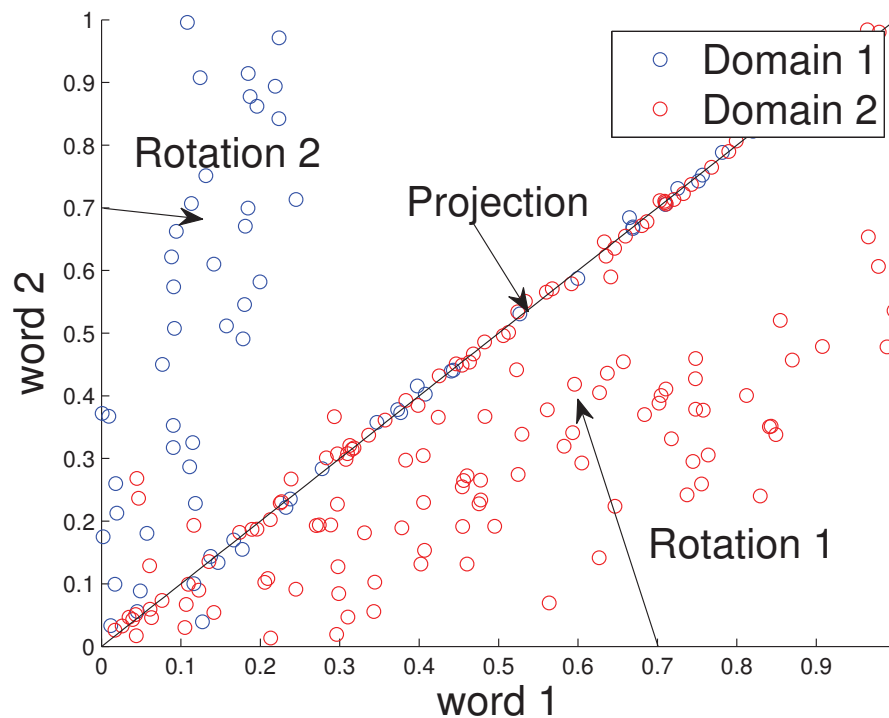


Figure 7.6.: Toy example for informativeness of the projections for variety linguistics: Word-Vectors with frequency values from text documents Domain 1 and Domain 2 are rotated into a common latent feature; the rotation magnitude represents how strongly the words need to be adapted to make the domains similar.

From the figure, we learn that military conflicts like the Israel-Lebanon war 2006 or the annexation of the Krim in the Ukraine by Russia 2015 are highly discussed in the Wikipedia talk pages. On the other hand, the news paper articles from the Die Zeit magazine primarily speak about the financial and European crisis. In the different document collections we have a clear dissimilarity in the usage of the word *Krise* (crisis). The Israel-Lebanon war is only in Wikipedia discussions a topic. Nonetheless, there are also similarities. The Euro crisis for instance is a topic in both collections.

Next, we perform the subspace based regularization to extract a subspace where both text collections follow a similar distribution as explained above. We sample documents from the Wikipedia talk pages and the Die Zeit magazine and perform SGD on the Stiefel manifold to minimize the optimization from Equation 7.1. The resulting projection matrix describes the projection onto a latent subspace spanned by the factors. To extract these factors we inspect the matrix and use the corresponding column vectors as the latent factors. Based on these factors, we calculate the ranking list of the top 10 words with respect to each factor as described in Section 3.1.1. In Table 7.2, the top ranked words are reported. Comparing the top words from the factors with the topics shown in Table 7.1, we see that the regularized factors cover main subjects from both document collections.

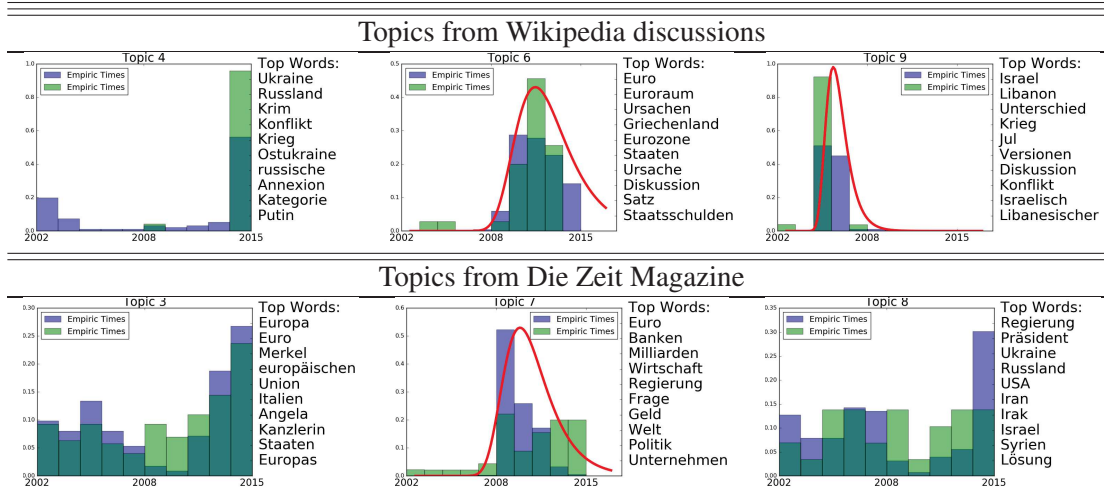


Table 7.1.: Three topics extracted from Wikipedia discussion pages and news article from the Die Zeit magazine that contain the word *Krise*. Top: In Wikipedia discussions the focus is on military crisis. Bottom: In the news article the financial crisis is more prominent.

"Factor 1"	"Factor 2"	"Factor 3"	"Factor 4"	"Factor 5"
"Geschichte"	"Israel"	"Euro"	"Ukraine"	"Banken"
"Ukraine"	"Libanon"	"Milliarden"	"Russland"	"Euro"
"Krim"	"Jul"	"Eurozone"	"Konflikt"	"Krieg"
"Google"	"Versionen"	"Euroraum"	"Krim"	"Meinung"
"Annexion"	"Unterschied"	"Griechenland"	"Ostukraine"	"Beitrag"
"Banken"	"Diskussion"	"Google"	"russische"	"Krim"
"Euro"	"Krieg"	"Waehrung"	"Kiew"	"Frage"
"Merkel"	"Israelisch"	"Laender"	"Euromaidan"	"Geld"
"klar"	"Libanesischer"	"Staaten"	"Kategorie"	"Geschichte"
"Regierung"	"Konflikt"	"Bank"	"ukrainischen"	"Problem"

Table 7.2.: Top ranked words for factors extracted from Wikipedia discussions and the Die Zeit news articles containing the word *Krise* (crisis) by a factor model with projection based regularization.

To qualitatively measure the projections for the distribution matching, we calculate the amount of rotation the projection induces in the vector space for each word. To amount of rotation is calculated by

$$\|P'w^i\|_2^2.$$

This is the length of the word w_i represented as Word-Vector w^i projected onto the latent subspace. See for example Figure 7.6. This value can be interpreted as the amount of rotation of Word-Vector w^i when applied to projection matrix P . The more rotation, the more important is

7. Use Case Variety Linguistics

"Max rot."	"Closest"	"2nd closest"	"3rd closest"
"Ukraine"	"Russland"	"Ostukraine"	"Kiew"
"Euro"	"Milliarden"	"Banken"	"Griechenland"
"Konflikt"	"Krieg"	"Bezeichnung"	"Ostukraine"
"Israel"	"Libanon"	"Jul"	"Versionen"
"Krieg"	"Konflikt"	"Bezeichnung"	"Annexion"
"Libanon"	"Israel"	"Jul"	"Versionen"
"Versionen"	"Jul"	"Unterschied"	"Libanon"
"Jul"	"Unterschied"	"Versionen"	"Israelisch"
"Krim"	"erklaert"	"Wort"	"Berlin"
"Unterschied"	"Jul"	"Versionen"	"Israelisch"

Table 7.3.: Words that face maximum rotation when projected onto the latent subspace extracted by a projection based factor model on Wikipedia discussion and Die Zeit articles. First column: Words most adapted to make Wikipedia discussions and Die Zeit articles similar in distribution. Columns 2-4: Closest words in the subspace spanned by the factors in terms of Euclidean distance.

this word to make the two document collections similar in distribution.

Further, we estimate the closest words in term of Euclidean distance in the latent subspace to the most important words. This can be used as a similarity measure:

$$\text{sim}(w_1, w_2) \propto \|P' \mathbf{w}^1 - P' \mathbf{w}^2\|_2^2.$$

In Table 7.3, we report the words that are most important for making the document collections more similar. In the first column, the words that are most rotated by the projection are reported. Here, military related words are most prominent. In the remaining column, we report those word that are most similar to the words that have been adapted the most by the projection.

Finally, we investigate how the two document collections get more similar in distribution in the subspace. In Figure 7.7, we plot how the projection matrix rotates words in terms of Word-Vectors.

7.5.2. Semantic Varieties in Social Media

Similar to the first experiment, we extract subspace based regularized factors that match the distributions of two collections of Amazon reviews. In Figure 7.8 we plot for two words the TF-IDF values in the vector space of the Word-Vectors from Amazon reviews about books (the source domain) and electronics (the target domain). The top figure shows the TF-IDF values that correspond to the words *professional* and *interesting* as Word-Vectors from both domains. The bottom figure shows the TF-IDF values that correspond to the words *display* and *author*. The Word-Vectors from the book reviews are represented by blue crosses and the Word-Vectors from the electronic reviews are plotted as red circles. In each figure, the left plot shows the Word-Vectors that correspond to the words before projection, and the right plot shows the Word-Vectors after projecting them with the projection matrix we found with the proposed method.

7.5. Qualitative Results

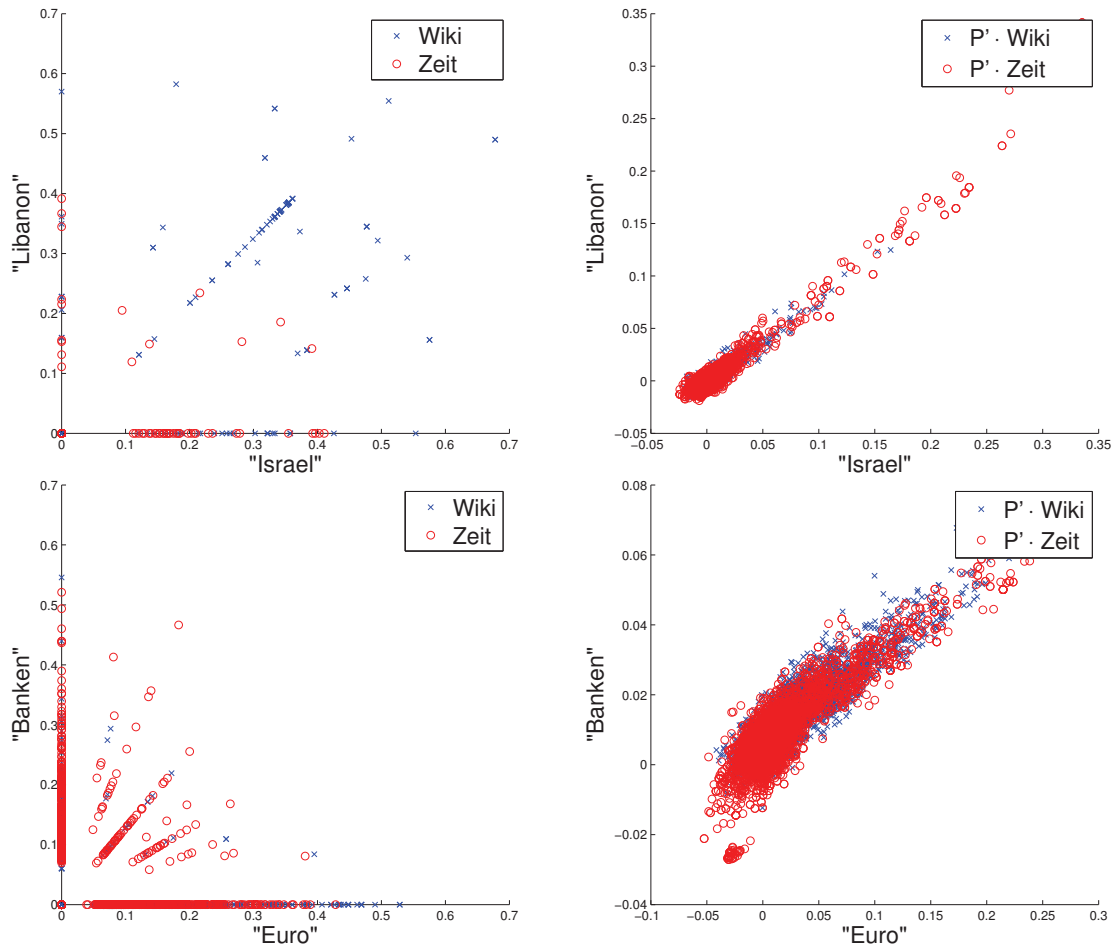


Figure 7.7.: Word-Vectors with frequency values in the original feature space (left) and projected space (right), for the adaptation of the words “Libanon” and “Israel” (top), and “Banken” and “Euro” (bottom).

We see that the words *professional* and *interesting* are important for the Domain Adaptation since the corresponding Word-Vectors are rotated in the vector space. The found projection matrix makes the corresponding components of the Word-Vector also more similar in the latent feature representation. This makes sense, since both words represent a common positive connotation; they are only differently distributed in the two original domains. On the other hand, the conceptually orthogonal words *display* and *author* are less important for Domain Adaptation: there is only little rotation of the Word-Vectors in the corresponding components. This backs up the hypothesis that the found projections help interpreting the adaptation needed to bridge the given domains of Word-Vectors.

To further investigate the informativeness of the projections learned for variety linguistics, we visualize the words in a 2-dimensional map. We use the method of Stochastic Neighborhood

7. Use Case Variety Linguistics

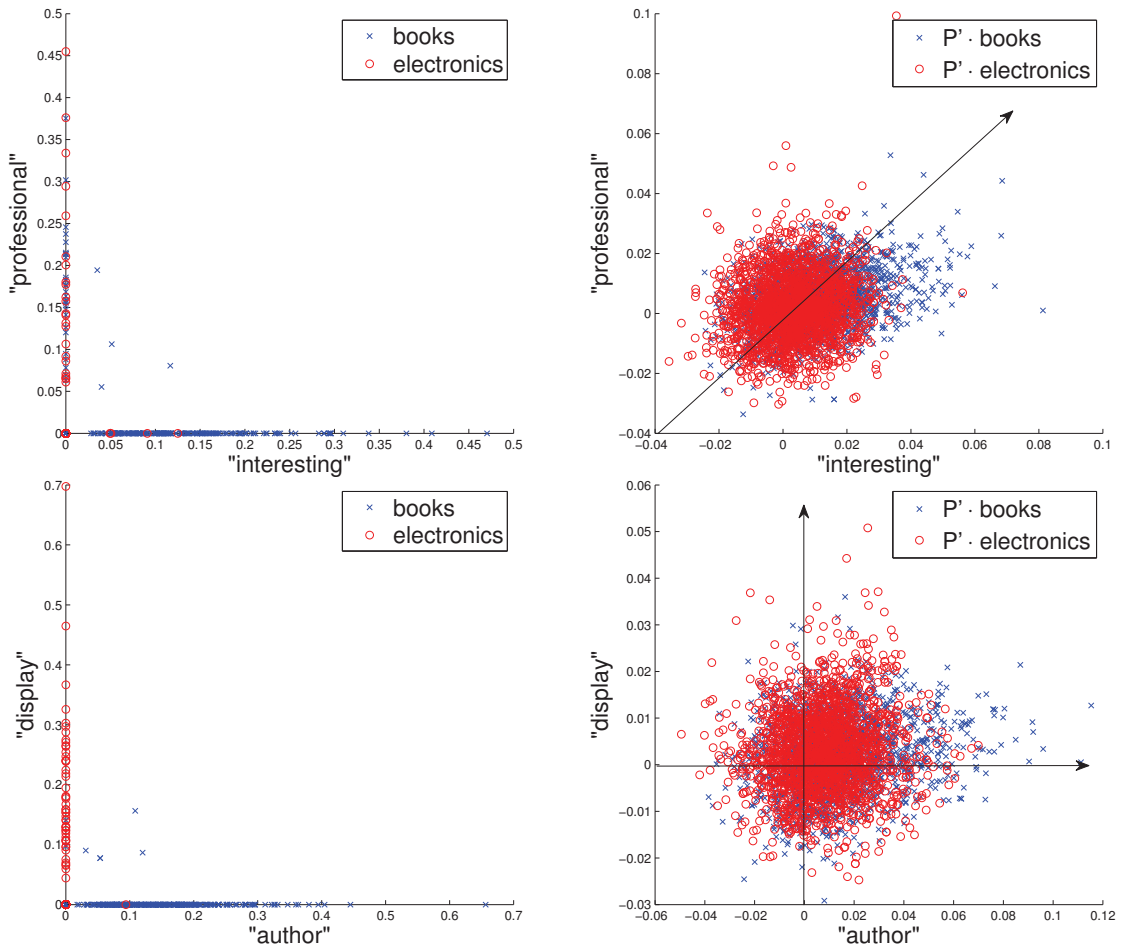


Figure 7.8.: Word-Vectors with frequency values in the original feature space (left) and projected space (right), for the adaptation of the words “professional” and “interesting” (top), and “display” and “author” (bottom).

Embedding (SNE) by [vdMH08]. This method models the joint probability of two words w_i, w_j as $p(w_i, w_j) \propto e^{-\|x_i - x_j\|^2}$ for x_i, x_j low-dimensional feature representations of the two words by SNE.

In Figure 7.9 we visualize positive adjectives before and after projection with the optimal projection matrix for Domain Adaptation in the same 2-dimensional space for reviews from books and electronic articles. The distances between the adjectives get smaller after projecting. For instance the words *perfect* and *useful* are much closer after projection compared to the original data. The word *perfect* appears in 54 reviews of books but in none of the reviews of electronic articles. The word *useful* appears in 106 reviews of electronic articles but only in 54 reviews of books. This distributional mismatch can be seen in the distance of the words in the original space. Clearly, the new feature representation in terms of the factors by the optimal

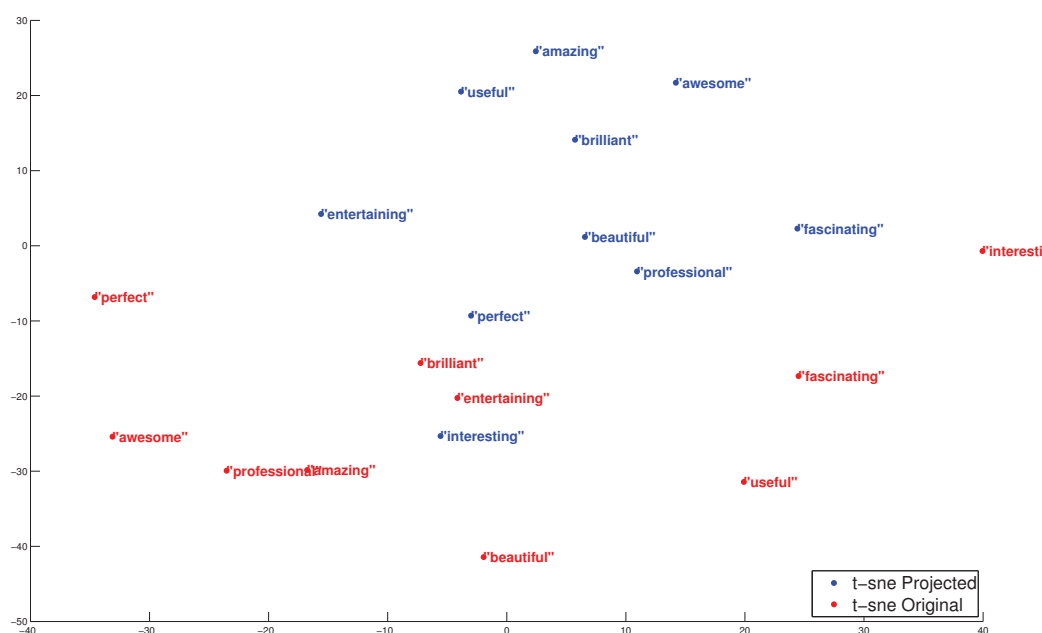


Figure 7.9.: Visualization of positive adjectives before and after projection in the space extracted by the method SNE.

projection matrix results in smaller Euclidean distance and hence larger joint probability of the two words.

7.6. Quantitative Results

To quantitatively evaluate the proposed factor models with regularization to match document collections, we estimate how well the subspace spanned by the factors transfer information from one collection to another in terms of a classification task. We quantify how good possible document labels from one document collections can be classified using another collection as training data. This is a Domain Adaptation task that evaluates the subspace spanned by the factors by a text classification quality. For example if we use reviews for electronic appliances with labels about their sentiment. We could estimate the quality of the extracted factors by a classifier for unlabeled review about DVDs for example. When the dissimilarity increases between texts from the labeled data set and the texts we want to classify, expected performance decreases. [BBC⁺10] showed that the expected error on a data set A of a classifier trained on a data set B correlates positively with the distributional difference between the data sets. The variety linguistics task is to find common distributional ground between data sets. The quantitative evaluation is done by estimating the quality of classifier trained on one data set and applying it on another.

To estimate the quality of the regularized factors in terms of Domain Adaptation, we estimate

7. Use Case Variety Linguistics

the quality of a document classification using the factor representation of the documents. For two document collections with difference distributions of the Word-Vectors and document labels like sentiments or time stamps, a subspace is extracted that is spanned by factors from the above mentioned regularized factor model with distribution matching. All Word-Vectors are projected into the corresponding factor representation and a classifier is trained on one collection and applied the another collection as test with N_{te} documents.

The prediction quality is estimated by accuracy:

$$accuracy = \frac{\#corr}{N_{te}},$$

for $\#corr$ the number of correctly classified documents from the test document collection. The accuracy calculates how good a classifier predicts the document labels.

We compare our proposed regularized factor model by SGD on the Stiefel manifold using exponential maps (Stiefel) and projections based on QR-Decomposition (Projection) with five state-of-the-art Domain Adaptation methods: covariate shift adaptation by Kernel Mean Matching (KMM) by [HSG⁺07a], Transfer Component Analysis (TCA) by [PTKY09], SGD on the Grassmann manifold (GrExp) by [BHLS13], Gradient Flow Kernel (GFK) by [Gra12] and Joint Distribution Adaptation (JCA) by [LWD⁺13].

All experiments were repeated several times; the reported accuracy values correspond to the smallest costs reached during the optimizations. The start points for the optimization are uniformly drawn from the Stiefel manifold. The SGD finds a projection matrix into a p -dimensional latent space in the vector space of the documents that is best suited for Domain Adaptation from the source to the target domain. For all experiments we set the dimension $q = 100$ for all methods and the weight λ to 5. These values have proven empirically to perform best overall data sets; additional, we show a sensitivity analysis on these two parameters. Unless stated otherwise, we let the SGD perform 1000 steps, after which all experiments showed convergence. We also investigate how the dimension q influences the quality of the Domain Adaptation for the subspace based methods. Although we get better results for higher dimensions on some data sets, the relative order of the methods based on Accuracy does not change.

We project all sampled documents onto the new feature representation, and train an SVM classifier on the source documents (after projection) and their labels only. Finally, we use labels for the target domain to estimate the Accuracy of the classifier on the target domain (after projection). The labels from the target domains are only used to estimate the performance on the target domain. We use an RBF kernel for the SVM with the meta parameter γ . The reported accuracies are the highest ones found by a grid search over the two parameters γ for the kernel and C for the misclassification penalty for the training of the SVM.

7.6.1. Linear Domain Adaptation

In the first experiment, we use documents belonging to one designated domain of given document collections as source domain and a different domain as target domain. For example, we use DVD reviews as source domain and book reviews as target domain. On the Amazon dataset, we experiment with all possible choices for source and target domain. On the Reuters and the 20 newsgroups dataset, we configure the target and source domains as explained above. We

Table 7.4.: Accuracies on the Amazon reviews, performing Domain Adaptation from one source domain to one target domain. $X \rightarrow Y$ means we train on reviews from X and test the classifier on reviews from Y .

	E→D	E→B	E→K	D→E	D→B	D→K
KMM	64.7	65.2	80.3	73.7	69.55	77.2
TCA	68.7	70.7	81.8	70.7	74.3	74.1
GrExp	61.8	61.8	66.2	58.2	66.0	58.8
GFK	59.8	59.3	68.2	59.4	56.3	61.2
JCA	71.0	67.2	80.8	71.6	76.6	75.4
Projection	75.0	73.7	77.2	67.6	71.7	71.2
Stiefel	75.2	75.0	81.4	75.0	78.9	76.2
	B→E	B→D	B→K	K→E	K→D	K→B
KMM	73.0	69.55	73.8	76.7	67.8	63.7
TCA	68.0	71.2	69.6	83.9	73.5	74.6
GrExp	57.0	59.6	59.2	62.2	60.4	60.4
GFK	60.4	58.5	61.7	66.2	62.7	60.5
JCA	70.8	73.8	75.7	77.4	71.0	62.6
Projection	66.0	71.5	68.2	79.8	78.5	74.1
Stiefel	73.4	78.1	76.8	83.3	78.9	76.2

perform SGD on the Stiefel manifold to get an optimal projection matrix. Here, we use both domains but no labels. Then, the reviews from both domains are projected into the new low-dimensional latent feature representation. A Support Vector Machine (SVM) is trained on the projected source domain reviews and applied on the projected target domain reviews.

In Tables 7.4 and 7.5 we report the results of the first experiment. The SGD on the Stiefel manifold results in a new feature representation for Domain Adaptation with the highest accuracies over all domains. KMM, TCA and GFK also show good results on some of the domains, but on average they deliver worse accuracies than SGD on the Stiefel manifold. On the Reuters dataset, Stiefel outperforms KMM and TCA. On the 20 newsgroups dataset, Stiefel outperforms TCA and GrExp. The optimization on the Grassmann manifold has the worst performance of all methods tested.

Comparing the projection and exponential map on the Stiefel manifold, we see differences on all data sets. On the Amazon dataset and the Reuters data set, the optimization with exponential map performs much better.

To investigate the quality of the SGD solution, we perform additional experiments. We compare SGD to standard gradient descent (GD1) with random starting points. Further, we use the optimal projection matrix P^* found by SGD as starting point for a gradient descent (GD2). The second setting serves to illustrate that the optimum found by SGD cannot improve much more. The rationale behind using SGD is, besides its applicability to large data sets, that the random behavior at the start of the SGD process makes it less prone to get stuck in local optima. While GD will stay in the first local optimum it finds, SGD still can escape the trap and end up in a possibly better local optimum. This is important, since our optimization problem is non-convex:

7. Use Case Variety Linguistics

Table 7.5.: Accuracies on the Reuters and 20 newsgroups datasets

	Reuters			
	C1→C2	C2→C1	C2→C3	C3→C2
KMM	60.1	56.8	58.5	56.2
TCA	53.0	51.5	58.1	55.8
GrExp	65.0	65.0	70.0	56.8
GFK	72.9	66.1	68.7	66.4
JCA	77.4	80.7	75.3	72.8
Projection	70.0	69.3	72.9	58.2
Stiefel	84.2	80.9	74.7	62.4

	20 newsgroups					
	Conf1	Conf2	Conf3	Conf4	Conf5	Conf6
KMM	96.8	84.4	98.4	91.2	98.5	95.3
TCA	94.4	87.7	96.1	90.1	94.0	88.9
GrExp	88.8	86.4	98.6	87.8	96.7	89.3
GFK	84.0	74.6	91.9	72.4	86.5	79.0
JCA	99.7	73.6	55.5	73.0	96.8	88.6
Projection	98.7	87.1	99.4	96.2	99.6	96.4
Stiefel	99.4	93.0	99.3	96.6	99.5	97.4

Table 7.6.: Minima reached by SGD and GD with random starting points (GD1), respectively starting from the SGD results (GD2).

SGD	E	D	B	K
E	0	0.0024364	0.0021104	0.0046920
D	0.0028555	0	0.0004567	0.0033506
B	0.0020263	0.0004198	0	0.0027398
K	0.0044783	0.0034033	0.0026004	0
GD1				
E	0	0.0031001	0.0033133	0.0025170
D	0.0028865	0	0.0012749	0.0034631
B	0.0026958	0.0015223	0	0.0034756
K	0.0024969	0.0034155	0.0035037	0
GD2				
E	0	0.0024360	0.0021101	0.0046920
D	0.0028553	0	0.0004560	0.0033503
B	0.0020254	0.0004194	0	0.0027393
K	0.0044782	0.0034032	0.0026000	0

while the MMD is convex in the Hilbert space induced by the corresponding kernel, it is not convex with respect to a projection matrix of the Word-Vectors. All experiments are repeated 10 times and the results presented are the lowest minimum found for the corresponding methods.

Table 7.7.: Maximum Mean Discrepancy (MMD) measure on the Amazon review category domains.

	E	D	B	K
E	0	0.0177	0.0207	0.0067
D	0.0177	0	0.0174	0.0173
B	0.0207	0.0174	0	0.0200
K	0.0067	0.0174	0.0200	0

In Table 7.6, we report the optimal values found by minimizing only the linearized MMD (see Equation 7.2.3) using the gradient methods. On the top of the table, we show the optima found by SGD using row X as source domain and column Y as target domain. The second table from top shows the optima found by gradient descent using random starting points (GD1). The table at the bottom shows the optima found by gradient descent using the result from SGD as starting point for optimization (GD2).

Comparing the different gradient methods, SGD finds always a better local optimum than GD1 except for the categories kitchen (K) and electronics (E). These two text collections are already similar in terms of MMD, as we will discuss in the next section. We assume that this closeness in distribution results in fewer local minima. When we start a standard gradient descent from the result found by SGD (GD2), we see that only rarely we can find an only slightly lower MMD (at the seventh position after decimal point).

Table 7.4 shows the accuracies on the target domains using documents from only one category as source domain. Choosing the right category might result in better performance. In the experiments on the Amazon reviews data, we find always one category that outperforms the other categories. For instance, for the categories kitchen (K) the best results are attained when we use the documents from the category electronics (E) as source domain. All other categories cannot bring equivalently good results when employed as source domain.

To investigate this behavior we calculate the Maximum Mean Discrepancy as defined in Equation (7.2) to estimate the difference of the distributions of the target and source domains. Table 7.14 shows the MMD values using documents from certain categories. For the category electronics (E), the documents from the category kitchen (K) are closest in distributions. Comparing this result with the accuracies in Table 7.4 on the target domain with documents from category electronics, the documents from category kitchen performs best for Domain Adaptation. The documents from reviews about DVDs (D) have similar MMD values among the other categories. This is also reflected in the accuracies above that show no clear category that performs best as source domain. Also, the category kitchen behaves similar to electronics, and books similar to DVDs.

Hence, employing prior knowledge of the target domain to choose the right source domain would be beneficial. Since in many cases this information might not be available, one could resort to using documents from a mixture of all categories but the one used as target domain. In the next experiment, we investigate this setting on the Amazon dataset. The documents from a designated category (E,D,B,K) are used as target domain. From this category we use only the documents. From the other categories we use documents and labels as source domain as in the

7. Use Case Variety Linguistics

Table 7.8.: Accuracies on the target domains using all the other categories as source domain. The column with label X corresponds to the Domain Adaptation task $(E \cup D \cup B \cup K \setminus X) \rightarrow X$

	E	D	B	K
KMM	81.0	75.2	72.5	83.9
TCA	81.4	77.8	74.7	84.9
GrExp	68.7	66.3	62.2	70.7
GFK	68.7	66.3	62.2	70.7
JCA	77.0	72.7	74.9	82.3
Projection	81.0	75.1	72.7	80.8
Stiefel	82.0	78.6	76.3	83.7

Table 7.9.: Accuracies on the target domains using all the other categories as source domain using cross validation for the optimal dimension parameter. The column with label X corresponds to the Domain Adaptation task $(E \cup D \cup B \cup K \setminus X) \rightarrow X$

	E	D	B	K
TCA	81.4	78.3	75.4	85.2
GrExp	68.7	66.3	62.2	70.7
GFK	81.0	77.5	76.3	82.7
Stiefel	82.3	78.4	77.0	85.3

experiments above. Since the source documents stem from three times as many categories as before, in this experiment we let the SGD run for three times as many steps.

In Table 7.13 we report the accuracies on the target domains for one category using all other categories as source domains. The overall performance on the subspace found by the optimization on the Stiefel manifold is better than KMM and TCA. Again, the optimization on the Grassmann manifold results in the worst results. Comparing the exponential maps to the projections, the computationally more expensive exponential maps find more optimal subspaces for Domain Adaptation. This shows that also on a mixture of different categories as source domain, Stiefel manifold optimization results in suitable projection matrices for Domain Adaptation.

Additional, we perform an experiment with cross validation for the dimensionality of the subspace for the methods: Transfer Component Analysis (TCA), SGD on the Grassmann manifold (GrExp), Gradient Flow Kernel (GFK) and our approach (Stiefel). We cut off 10% of the target data to find the optimal dimensionality by maximizing the accuracy. On the remaining data, we calculate the final accuracies. The results are reported in Table 7.9. The SGD on the Stiefel manifold results in the highest accuracies, TCA and GFK perform slightly worse.

Convergence

The advantage of SGD directly on the Stiefel manifold is that we avoid additional projection steps after each SGD step to satisfy the orthogonality constraint of the matrices. This additional

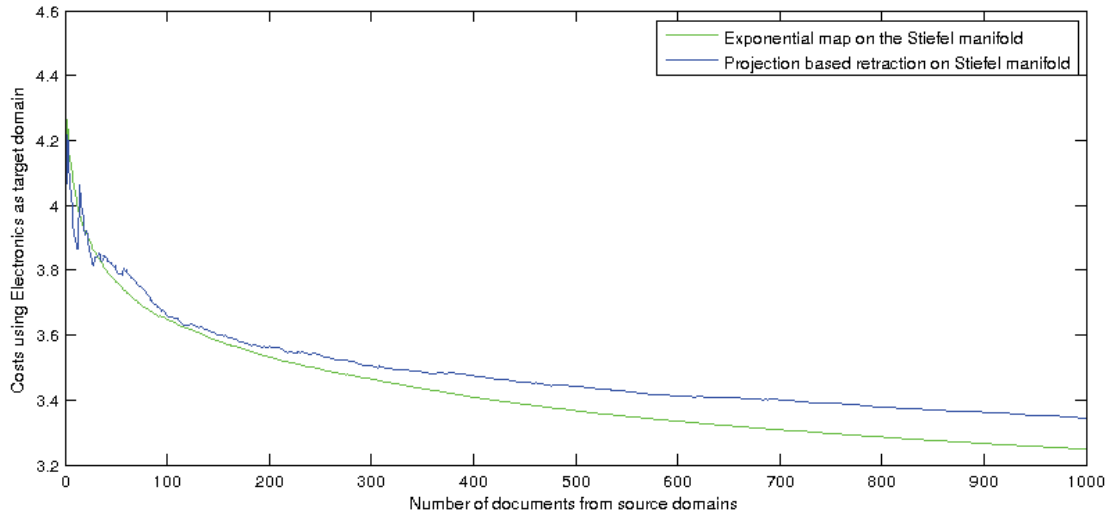


Figure 7.10.: Convergence of the costs from the optimization problem after a number of documents have been seen. As target domain we use electronic reviews and the source domain consists of the kitchen reviews. On all other possible settings of target and source domains, we get similar convergence results.

step will induce errors after each SGD step. Consequently, we expect slower convergence when we perform only projections onto the Stiefel manifold.

Next, we investigate the convergence of the stochastic gradient descent on the Stiefel manifold. We show the costs of the optimization function for the target domain of electronic reviews. As source domain we use reviews about kitchens. Figure 7.10 plots these costs depending on the number of documents from both the target and source domain for the optimization. We report the course of the costs during the optimization of the Stiefel manifold using both a projection by an QR-decomposition onto the manifold and the exponential map that moves along the manifold.

Figure 7.10 shows a fast convergence for both methods. The exponential map has a faster convergence than the projection method, from having seen only few documents onwards. The convergence is quite stable for both methods. The optimization with exponential maps reaches lower costs than the optimization with the projection. This shows that exponential maps can indeed result in better optimization performance using the proposed cost function: optimization on the Stiefel manifold with exponential maps converges faster and reaches a lower cost. This matches the results from the previous experiment that showed better performance on some domains when using exponential maps compared to projections.

Parameter Sensitivity Analysis

The proposed optimization method depends on the dimension of the latent feature representation and the regularization parameter in the cost function. While in the main experiments we used fixed values for the dimension and the regularization parameter, here we investigate different values in a sensitivity analysis.

7. Use Case Variety Linguistics

dimensionality q	E	D	B	K	C1→C2	C2→C1	C2→C3	C3→C1
40	77.0	75.9	73.3	79.9	76.5	70.7	66.8	59.0
60	72.2	66.3	70.8	75.4	75.7	70.6	65.3	58.6
80	76.3	74.1	72.2	78.3	73.2	72.3	69.8	58.1
100	74.8	73.5	72.4	80.0	72.6	72.4	72.9	58.2

dimensionality q	Conf1	Conf2	Conf3	Conf4	Conf5	Conf6
40	97.7	92.5	99.9	97.3	99.2	97.0
60	99.6	92.0	99.6	97.7	99.5	98.2
80	99.2	90.8	99.7	96.3	99.7	97.4
100	99.4	91.7	99.5	95.8	99.6	98.2

Table 7.10.: Accuracies on the projected target domain onto subspaces of various dimensions q for the target domains. The optimization is on the Stiefel manifold. The classifier is trained on the source domains projected onto the corresponding subspace. The first four columns with label X corresponds to the Domain Adaptation task on Amazon reviews ($E \cup D \cup B \cup K \setminus X \rightarrow X$); the next four columns correspond to the domain adaptation task on Reuters; the last six columns correspond to the domain adaptation task on the 20 newsgroups dataset.

The dimensionality of the latent feature representation and hence the used manifold $M(q, p)$ is a meta parameter that has to be chosen beforehand. It is clear that for a good performance we need a large enough number of dimensions to capture all necessary information. On the other hand, the higher the dimensionality, the more computation is needed to estimate the gradient steps. Beside this, too high-dimensional representations might introduce too much variance from the different domains. In Table 7.10 we show the accuracies on the target domains in the feature representations from the projection matrices found by stochastic gradient descent on the Stiefel manifold for various dimensions q . The results show that higher numbers of dimensions generally but not consistently correspond to slightly better accuracies. Hence, without labels for the target domain, the choice should be in favor of large dimensions. In case we have labels for the target domain, we can perform cross validation to find the optimal parameter q .

In the experiments so far, we used the maximum mean discrepancy and the regularization on the norm for the optimization with a fixed parameter $\lambda = 5$. Here, we analyze the difference of the Accuracy from the projections that have been found by stochastic gradient descent with various weights on the regularization of the norm. Table 7.11 shows the accuracies for various weights λ . We see that the regularization of the norm is vital for the performance of the domain adaptation. Without the regularization, the found projection is not able to capture enough information from the domains for a good classifier on the target domain. Higher weights result in better performance on average. This means, that the regularization on the norm helps keeping enough necessary information from the domains to train a good classifier for the target domain.

weights λ	E	D	B	K	C1→C2	C2→C1	C2→C3	C3→C1
0	64.7	62.3	62.2	64.4	76.5	70.7	66.8	59
1	79.1	71.8	70.9	80.8	74.9	70.8	69.9	58.4
4	78.9	73.2	73.9	82.3	72.4	71.6	71.3	58.8
5	79.4	73.6	74.6	81.7	75.7	70.8	71.0	59.4
10	78.9	73.6	72.8	81.5	73.4	70.6	71.1	58.8
weights λ	Conf1	Conf2	Conf3	Conf4	Conf5	Conf6		
0	92.4	82.2	98.8	78.3	94.0	87.7		
1	99.3	89.9	99.5	97.4	99.5	98.5		
4	99.1	89.4	99.4	96.6	99.5	96.4		
5	99.7	87.9	98.9	96.5	99.7	96.4		
10	98.8	86.8	99.5	97.1	99.4	97.0		

Table 7.11.: Accuracies on the projected target domain onto subspaces with various weights λ in the optimization problem. The optimization is on the Stiefel manifold. The classifier is trained on the source domains projected onto the corresponding subspace. The first four columns with label X corresponds to the Domain Adaptation task on Amazon reviews ($E \cup D \cup B \cup K \setminus X \rightarrow X$); the next four columns correspond to the Domain Adaptation task on Reuters; the last six columns correspond to the Domain Adaptation task on the 20 newsgroups dataset.

7.6.2. Non-linear Domain Adaptation

So far, we extract only linear factors. Next, we use kernel methods to extract subspaces in an RKHS that are spanned by non-linear factors. The quality of these factors are again estimated by the classification quality in a Domain Adaptation task,

For the non-linear subspace for Domain Adaptation, we extract the first 100 principle components from the kernel matrix K for all samples from the sampled source domain data and the target domain. This means, for each $\mathbf{w}_{d_i}, \mathbf{w}_{d_j} \in \{T \cup S'\}$ we have $K = (k(\mathbf{w}_{d_i}, \mathbf{w}_{d_j}))_{i,j}$. We project all data samples (all source and training data) onto the subspace spanned by the extracted components and train a classifier on the source domain in this subspace. Next, we apply this classifier on the target domain in the subspace. We compare the sampling strategies without and with random features (Sampling, Sampling+RF) with Transfer Component Analyses (TCA) [PTKY11], Kernel Mean Matching (KMM) [HSG⁺07b], Gradient Flow Kernel (GFK) [Gra12] and Nyström sampling (Nyström) [KMT12] that also uses random samples. For TCA we also use 100 components. We use Gaussian kernels with optimized width parameter σ . For the classification we train an SVM with optimized error weight. For the random features, the results are mean values over 10 runs with random features of dimension 10.000.

The method by [GGS13] has the same objective as our sampling methods. They find those source domain points that minimize the MMD to the target domain. Compared to our method, the points are extracted by solving a quadratic optimization problem with constraints. This is computationally challenging when we have large source domains. Moreover, they do not directly select the points, they propose to learn weights of the points and remove those points that have

7. Use Case Variety Linguistics

Method	org vs. places	places vs. org	places vs. peo- ple	people vs. places	comp vs. rec	comp vs. sci	comp vs. talk	rec vs. sci	rec vs. talk	sci vs. talk
KMM	60.1	56.8	58.5	56.2	96.9	84.4	98.5	91.2	98.5	95.4
TCA	85.4	80.5	76.5	76.5	94.5	87.8	96.2	90.2	94.1	88.9
GFK	72.9	66.1	68.7	66.4	84.1	74.7	91.9	72.5	86.6	79.0
Nyström	79	79.9	72.2	67.6	98.7	88.7	98.9	94.7	99	96.7
Sample	90	82	83.5	79.2	99.1	92	99.2	98.3	99	96.2
Sample + RF	84.7	82.9	85.5	77.3	98	88.4	98.7	91.7	98	93.7

Table 7.12.: Accuracies on the Reuters and 20 newsgroups datasets. We compare our proposed greedy sampling methods (without and with random features) and projection with Kernel Mean Matching (KMM) and Transfer Component Analysis (TCA), Gradient Flow Kernel (GFK) and Nyström sampling (Nyström).

Method	$\{D \cup B \cup K\} \rightarrow E$	$\{E \cup B \cup K\} \rightarrow D$	$\{E \cup D \cup K\} \rightarrow B$	$\{E \cup D \cup B\} \rightarrow K$
KMM	81.0	75.2	72.5	83.9
TCA	81.4	77.8	74.7	84.9
GFK	68.7	66.3	62.2	70.7
Nyström	79.3	77.3	75.2	82.8
Sampling	82.4	79.1	77.2	85.2
Sample+RF	81.3	79.7	77.6	84.8

Table 7.13.: Accuracies on Amazon reviews using one product as target domains and all the other domains as source domain. We compare our proposed greedy sampling methods (without and with random features) and projection with Kernel Mean Matching (KMM) and Transfer Component Analysis (TCA), Gradient Flow Kernel (GFK), the Landmark method (LM) with projection and Nyström sampling (Nyström) for Domain Adaptation.

weights below a threshold. This threshold has to be chosen by hand. In the experiments we use the same threshold as they have done in their experiments.

The results of the first experiment are shown in Tables 7.12 and 7.13. The projections onto the components result in the best performances for all the domains. The subspace obviously covers the important invariant parts of the data very well. Using random features to approximate the kernel values results in the second best accuracies compared to the other methods.

Next, we explore how many source domain points have been chosen from which domain. Figure 7.11 shows histograms of the selected data points from the source domain for the different methods. The sampling strategy without and with random features and the GFK method uses a similar amount of samples from the source domains. The histograms show that for each target domain the methods have always one domain in the mixture of source domain where most of the samples are drawn from. For sampling, there is always one clear domain from which the method samples most from.

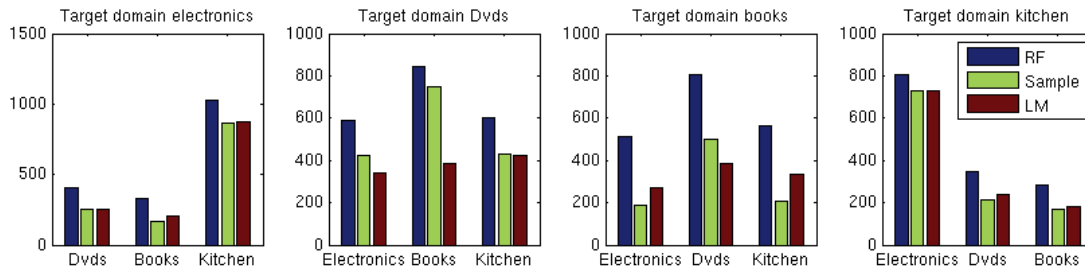


Figure 7.11.: Histograms of the selected points from Amazon reviews by the different sample strategy.

MMD	E	D	B	K
E	0	0.0177	0.0207	0.0067
D	0.0177	0	0.0174	0.0173
B	0.0207	0.0174	0	0.0200
K	0.0067	0.0173	0.0200	0

Table 7.14.: Maximum Mean Discrepancy (MMD) measure on the different domains from the categories from the Amazon reviews.

To investigate this further we calculate the Maximum Mean Discrepancy as defined in Equation 7.2 to estimate the difference of the distributions of the target and source domains. Table 7.14 shows the MMD values using reviews from the domains. For the electronics reviews (E), the reviews about kitchens (K) are closest in distributions. Comparing this result with the accuracies from above, on the target domain with reviews about electronics, source domain kitchen performs best for Domain Adaptation. Similar results can be seen for the other domains. Comparing the MMD of the domains with the sampled points from the last experiments, we see that the sampling method chooses the source domain points that results in low MMD best.

Finally, we investigate the influence of the random features on the quality of the Domain Adaptation. We perform several runs using different feature sizes. The plots in Figure 7.12 show a fast convergence already after some thousand random features. Experiments with random features of dimension less than one thousand has let to poor performance. This might be due to the slower convergence of the kernel matrix to the matrix of the inner products of the random features in the norm. In the future we will investigate this further.

7.7. Conclusion

We interpret variety linguistics as Domain Adaptation tasks and propose to use SGD on Stiefel manifolds to find a projection onto a latent subspace that is best suited to cover similarities across text collections. We provide update rules that compel the SGD steps to remain on the Stiefel manifold, and solve an optimization problem employing these steps. Since the Stiefel manifold encompasses projection matrices on Word-Vectors, the results are interpretable: the

7. Use Case Variety Linguistics

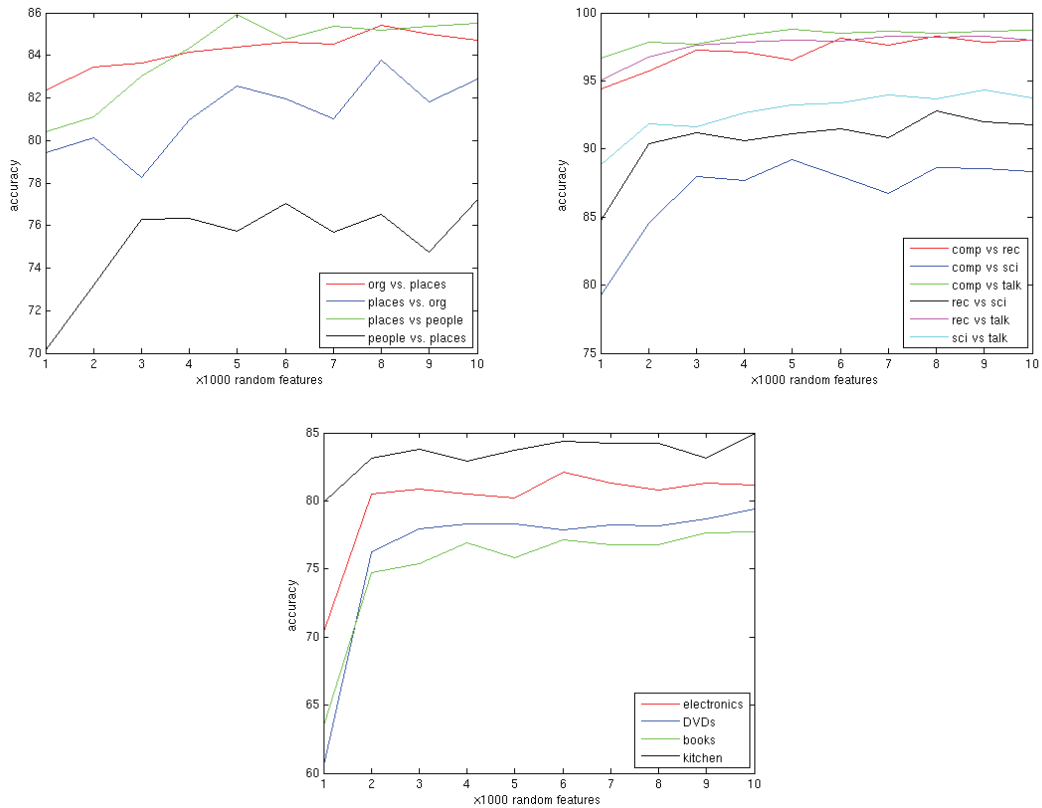


Figure 7.12.: The classification accuracies using different numbers of random features. Left: the Reuters dataset; right: 20 newsgroups dataset; bottom: Amazon reviews

importance of a word towards making the Domain Adaptation can be estimated by measuring the rotation magnitude of the projection of that word, as is illustrated by Figure 7.8. Furthermore, we have seen that in terms of quantitative evaluation, the proposed method (Stiefel method) performs at least as good as or better than competing state-of-the-art Domain Adaptation methods; optimization on the Grassmann manifold cannot compete (cf. Table 7.4). Kernel Mean Matching and Transfer Component Analysis can deliver comparable accuracies, but these methods are regularly outperformed by the Stiefel method as well (cf. Table 7.5). When increasing the amount of domains from which source documents are taken, this behavior remains (cf. Table 7.13): accuracy of the Stiefel method is typically best or equivalent to best, while every competing method performs sometimes equivalently and sometimes substantially worse. For variety linguistics, the Stiefel method delivers interpretable results without substantial loss, and even regularly to the benefit of accuracy. For variety linguistics with non-linear factor models, we propose a selection strategy on samples from a source domain that are best suited for Domain Adaptation to a target domain with a different data distribution. The samples are selected to keep the structure of the target domain points while adding some structure from the source domain points. Projecting onto the subspace of the selected samples and the target samples results in

7.7. Conclusion

a subspace that is well suited for Domain Adaptation from the source to the target domain. To apply this approach also on large scale data sets, we use random features to approximate kernel values. On large digital corpora, we show that our method performs well on Domain Adaptation tasks and variety linguistic tasks.

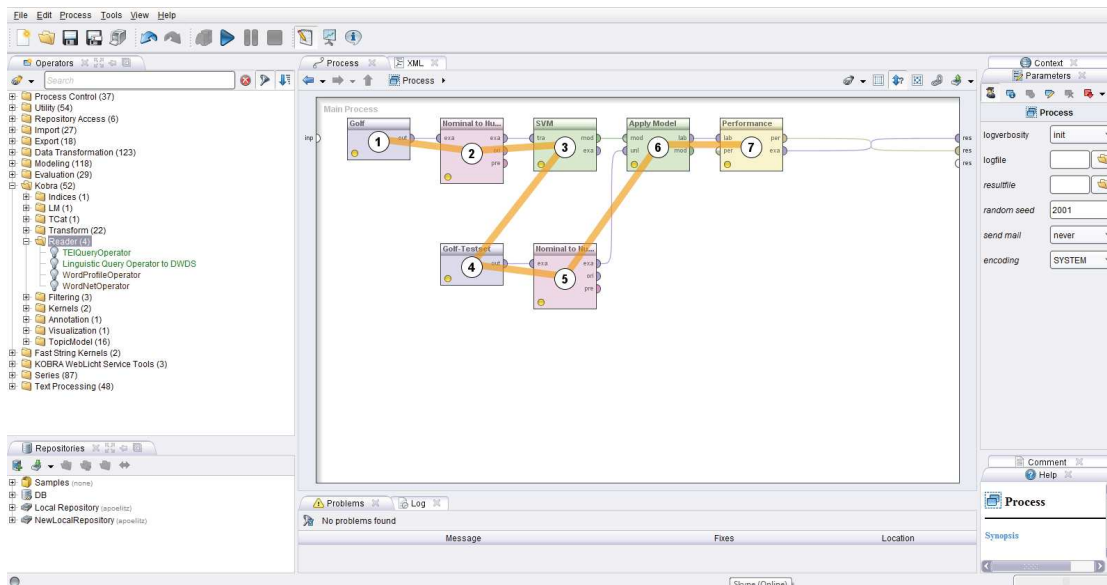


Figure 8.1.: RapidMiner user interface. Left: Operators for data loading, pre-processing, Data Mining methods, post-processing and data export. In the middle: Data Mining process. Right: Properties and parameters of the process and the operators.

8. Software and Integration

In this chapter, we describe the software developed in the thesis. All methods that have been used for the use cases are implemented in Java or Matlab. The code is publicly available on the web page <http://sfb876.tu-dortmund.de/auto?self=Software>. In the next section, we describe in detail the software and how it can be used to reproduce the use cases. The software is used in corpus linguistic research and teaching at the TU Dortmund University and the Mannheim University. Additionally, the methods have been integrated into modern language resources that are used for linguistic studies. We start the software description with an introduction to the Data Mining tool RapidMiner. RapidMiner is used and extended for corpus linguistic tasks.

8.1. RapidMiner

The RapidMiner [HK13] is a Data Mining toolbox used to perform data analysis on different data sources. RapidMiner offers the classical analysis and Data Mining steps from data retrieval to data transformation and pre-processing, performance of analysis and Data Mining methods

8. Software and Integration

to evaluation methods, post-processing and visualization. Individual processing steps are performed by so called **Operators**. The standard operators are separated into several categories and are organized in an ontology represented as folder structure in the operator explorer view on the left of the main screen as seen in Figure 8.1. The main categories of operators are:

- import/export operators: reading and writing of data
- data transformation operators: pre- and post-processing of data
- modeling: analytic and data mining methods on data
- evaluation operators: quality estimation of the modeling results.

The operators are compiled to a sequence of steps summarized in a so called **Process**. This process defines a flow of input data to processing operators that output result data. In the middle of the figure, an example process is shown with the execution order of the individual operators. Starting with reading data as CSV-file, the data is pre-processed by transforming nominal to numeric data. The modeling operator **SVM** builds a classification model that is applied on test data additional read in. Finally, the **Performance** operator is used to evaluate the model by standard measures. The operators have a number of parameters to be specified. On the right of the figure, the **Parameters** panel is shown as input mask for all parameters. Clicking on an operator, this panel shows the parameters that need to be set for this operator. Additional, a description of each operator can be found on the **Help** panel. A general introduction into Data Mining with RapidMiner can be found in the book [Nor12] by Matthew North.

8.2. Corpus Linguistics Plugin

The RapidMiner offers a convenient interface and a plethora of available analyses methods. Compared to low level interfaces and libraries for different programming languages, RapidMiner offers a more user friendly tool box. This makes the introduction of our methods more easy for linguistic researchers with little knowledge in computer science. We implemented the proposed latent variable methods as a plugin for the RapidMiner. For the different variants of LDA, different operators are available. Besides standard LDA with Gibbs sampling and Variational Inference, supervised versions with Gaussian, Beta, Uniform and Gompertz distributed document labels can be used for diachronic linguistic tasks. An implementation of LDA with word features and word groups via special Laplace and Group-Sparsity inducing priors is available to integrate word informations. Some of the latent factor methods can be generated with existing operators already available in RapidMiner. For example for LSA, the available operator for a SVD can be used. For variety linguistic tasks, we provide an operator that extract latent factors that match distributions of different document collections.

Additional to the latent variable methods, we also implemented a number of interfaces to the language resources. To access the different corpora, operators to execute linguistic queries on the different corpora at the Berlin Brandenburger Academia of Science are available. Besides the standard corpora, we also provide access to the dictionaries and the GermaNet (the German version of WordNet). To access the Wikipedia corpora, a TEI-reader is implemented that extends a

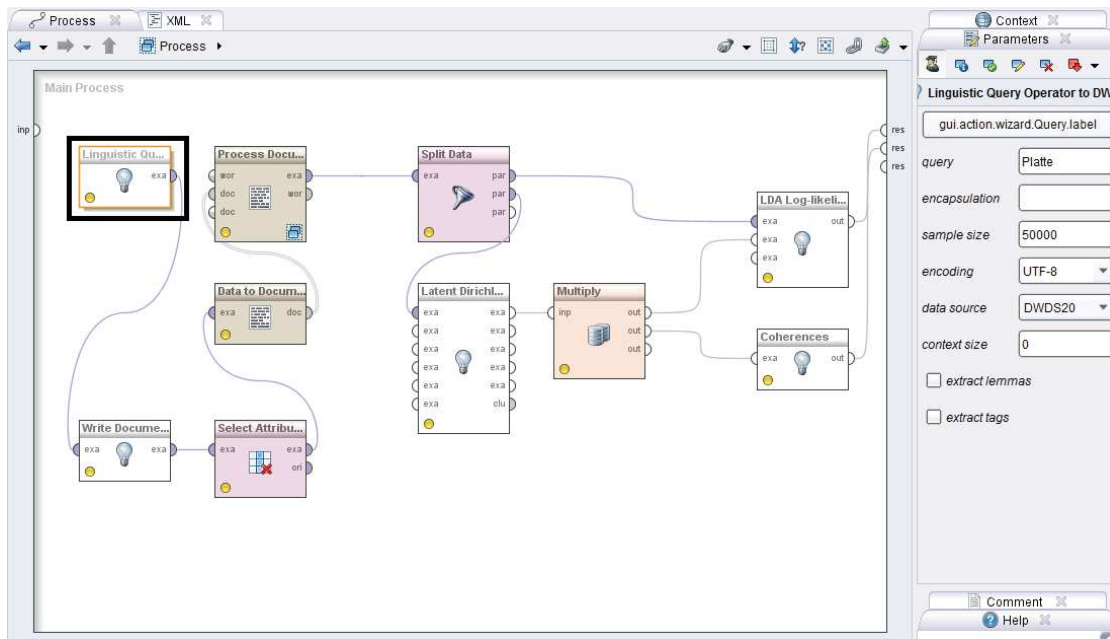


Figure 8.2.: Linguistic Query Operator as first step in a process to perform a linguistic task by latent variable methods. For a given query, we retrieve KWIC-lists from a corpus.

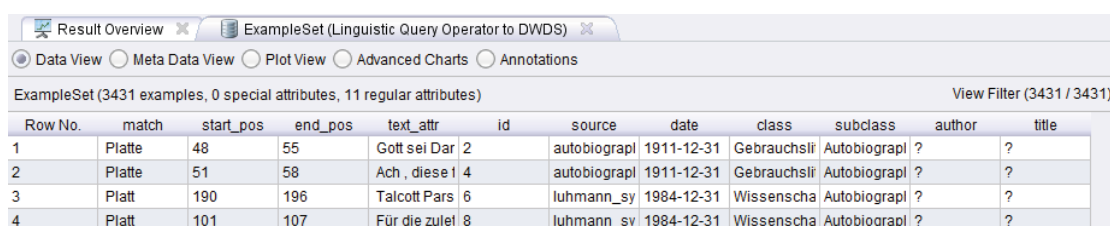
standard XML-stream reader to process the Text Encoding Initiative (TEI) tags, see [BEG⁺12]. Finally, preprocessing operators provide methods for text transformations and text visualization. In the next subsections, concrete examples for the use of the plugin are described. A reference for the individual operators is given in the appendix.

8.2.1. Interface to Linguistic Resources

The first step to perform linguistic tasks with the Corpus Linguistics Plugin is the retrieval of the data. The KWIC-lists or documents are extracted and internally represented as string. Standard text documents can be opened by the **Read CSV** operator from RapidMiner. For the linguistic corpora we implemented the Linguistic Query Operator as shown in Figure 8.2. The **Linguistic Query Operator** provides access to the DWDS Core-Corpus of the 20th century, the Core Corpus of the German text archive and the Die Zeit corpus of news articles from 1947 to 2014. For a given linguistic query, the operator retrieves a number of concordances and generates an example set that contains the texts, a time stamp and additional information about author and source. The query is sent to a server at the Berlin Brandenburger Academia of Science and a Perl script runs the query against a Dialing and DWDS Concordance (DDC) data base containing the corpora, see [Sok03]. The KWIC-lists are returned as JSON¹ files and the operator parses these information and generates the results. Depending on the corpus additional information about the genre of the corresponding documents are also available. Additionally, the position of the query

¹<http://www.json.org/>

8. Software and Integration



Row No.	match	start_pos	end_pos	text_attr	id	source	date	class	subclass	author	title
1	Platte	48	55	Gott sei Dar	2	autobiograpl	1911-12-31	Gebrauchslit	Autobiograpl	?	?
2	Platte	51	58	Ach , diese l	4	autobiograpl	1911-12-31	Gebrauchslit	Autobiograpl	?	?
3	Platt	190	196	Talcott Pars	6	luhmann_sy	1984-12-31	Wissenscha	Autobiograpl	?	?
4	Platt	101	107	Für die zulei	8	luhmann_sy	1984-12-31	Wissenscha	Autobiograpl	?	?

Figure 8.3.: Result example set from Linguistic Query Operator for a linguistic query. For each match of the query, we have an example with information about the match.

match in the retrieved snippet is given to efficiently identify to match. In Figure 8.4, we show the resulting example set from the Linguistic Query Operator.

For corpora and documents in TEI format, the **TEI Query Operator** provides a stream reader to process large files. Since these files are not indexed as the corpora from the Dictionary of the German Language, we cannot pose linguistic queries. Instead, standard regular expressions can be queried. For the main TEI formatted corpora, the Wikipedia articles and talk pages, the operator retrieves matches of the regular expressions on sentence, paragraph or postings level. These levels are semi-automatic annotated, see [ML14]. The operator itself implements an XML based stream reader to iterate over the elements from the TEI file. The resulting example set has the same schema as the example set from the Linguistic Query Operator.

To efficiently inspect the retrieved KWIC-lists, the **Annotation Operator** visualizes the text snippets and highlights the matches in the texts. We can also add additional labels or attributes to the texts to further annotated them. The operator generates a result as example set containing the texts of the snippets and the additional annotations.

For the retrieval of information from the additional language resources like dictionaries and WordNets, we implemented operators that can extract these information from local files (WordNet for instance) and retrieve them from the Dictionary of the German Language. The **WordNet Operator** takes a word as parameter and extracts similar words from an existing WordNet instance, given the path to the index, hyponyms and hypernyms for the data. The **GermaNet Operator** works the same, but uses the GermaNet source provided by the Seminar für Sprachwissenschaften at University Tübingen. The retrieval of these information is done by a web service at the Berlin Brandenburg Academia of Science via JSON files. The resulting example set contains for the word of interest given as parameter, the hyponyms and hyperonyms with additional examples and descriptions. Further, the **WordProfiles Operator** retrieves the word profiles provided by the Dictionary of the German Language. These profiles contain words that co-occur with a given word of interest and gives information about the relation between them, see [DG13].

8.2.2. Text Processing

Before we can use the KWIC-lists for latent variable methods, we need to generate Word-Vectors. With the **Text Processing Plugin** as provided by RapidMiner, text (general strings) can be transformed into a Bag-of-Words and Word-Vectors. Given text in an example set, an

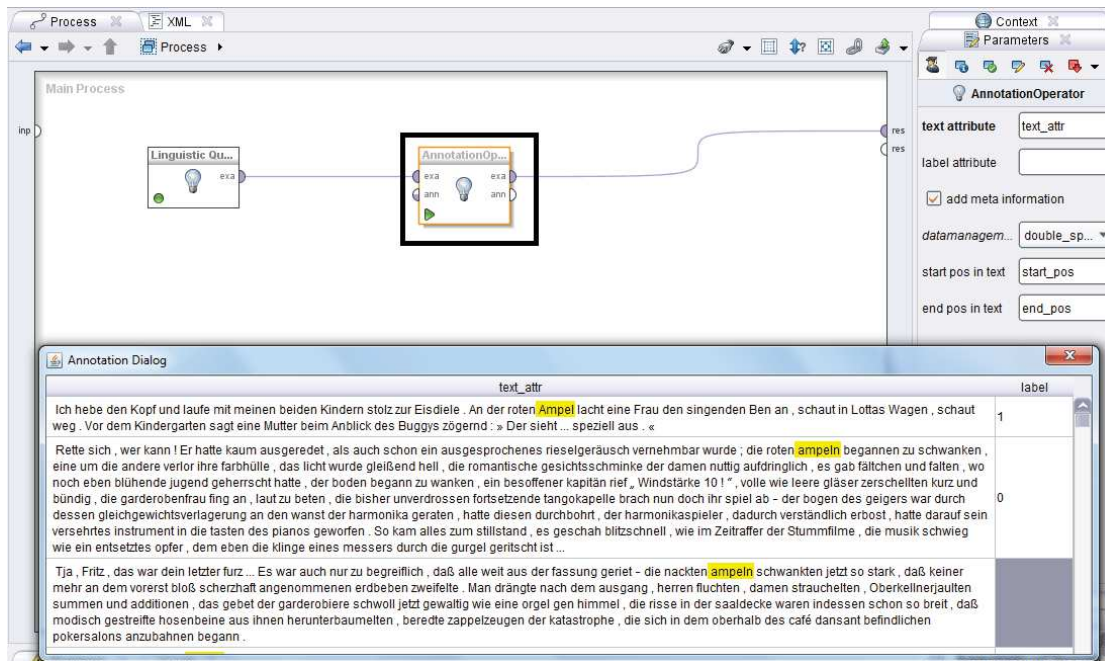


Figure 8.4.: Annotation Operator and annotation environment. Given an example set from TEI or Linguistic Query Operator, the snippets are visualized and the match is highlighted to inspect the results. Additional annotation can be added like a label.

internal data structure to represent the text is a generated. This data structure is called a **Document**. These documents are further transformed into Word-Vectors containing word occurrences and possible weights as TF-IDF. The text processing plugin offers additional methods to tokenize text by regular expressions or identification of words. Filtering operators can be used to filter out stop words, large tokens or tokens with no characters. Additional pruning mechanisms can be used to filter out words that appear in too many or too few documents. In Figure 8.5, we show how the snippets are transformed into Documents by the **Data to Documents Operator** and how we further generate Word-Vectors by the **Process Documents Operator**. Using the Process Document Operator, we can use different tokenizers to separate the text into tokens and prune words. The resulting example set contains each Document as Word-Vector in a table. In the Word-Vectors there can be pure occurrence information or weighted values like TF or TF-IDF values. We can choose between different methods to prune words. We can prune words with a frequency high or lower threshold, that appear more often or less than a given number or are below and higher a given rank.

8.2.3. Latent Topic Models

From the Bag-of-Words representation of the documents, we can use the sequences of word tokens for the extraction of topics by LDA. Our **Latent Dirichlet Allocation** operator takes the texts as Word-Vector with pure word occurrences and extracts latent topics by Gibbs sampling.

8. Software and Integration

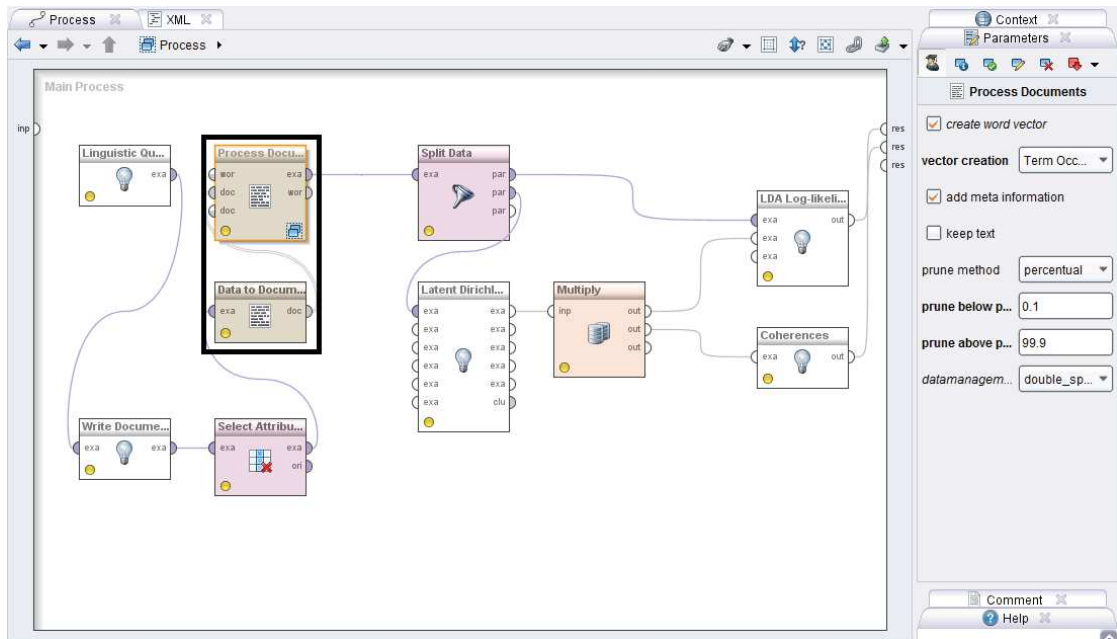


Figure 8.5.: Generation of Word-Vectors from example sets with a text attribute.

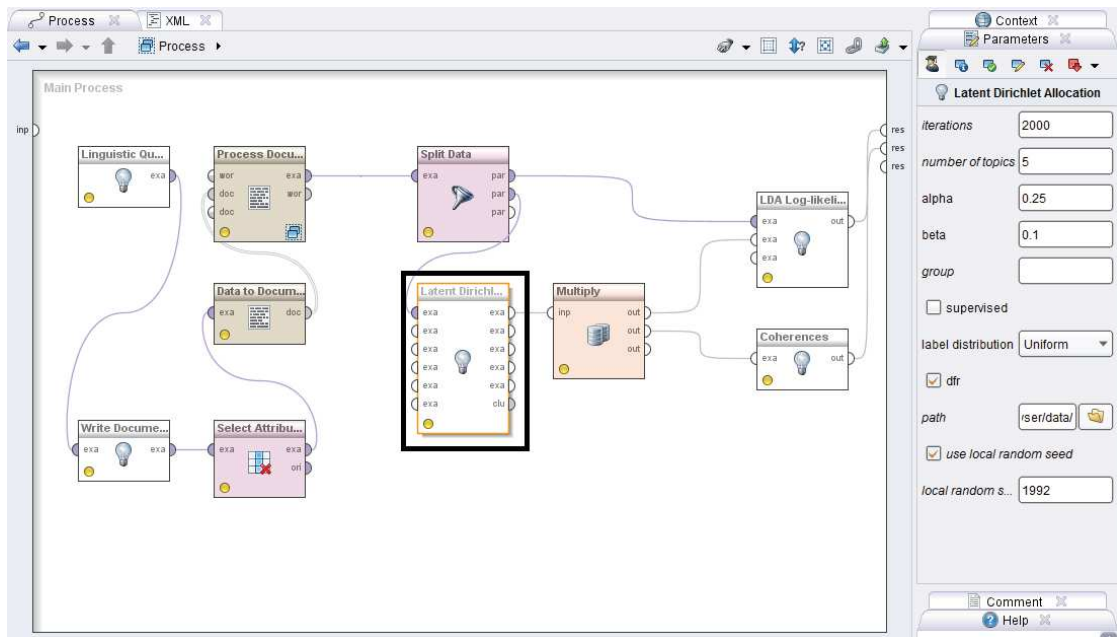


Figure 8.6.: Latent Dirichlet Allocation operator to extract latent topics from a document collection given as Bag-of-Words.

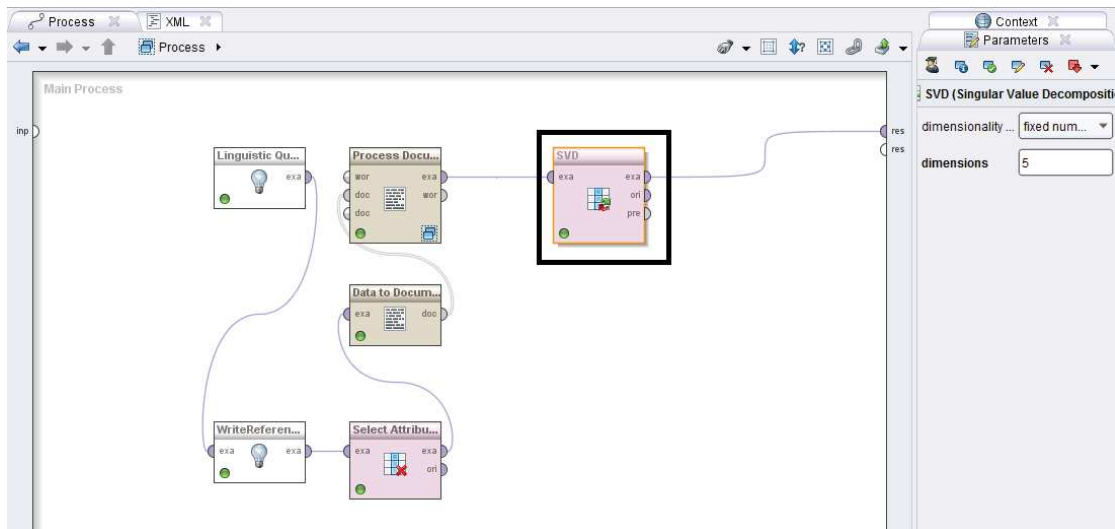


Figure 8.7.: Extraction of latent factors via Singular Value Decomposition operator from Rapid-Miner.

In addition, an operator that performs Variational Information for LDA is also available. Although we implemented both variants, in our experiments we used Gibbs sampling that showed good performance. For the Gibbs sampler, we need to specify how many iterations we want to process. Further, the number of topics to be extracted and the meta parameters of the Dirichlet priors need to be specified. For standard LDA, we un-check the supervised check box. For sLDA like temporal topic modeling, we check the supervised check box and specify the label distribution (Uniform, Beta, Gompertz). If we use sLDA, the input example set must contain a label attribute additional to the Word-Vectors. For visualization of the results we check the **df** check box and specify the path to the data folder for the DFR-Browser. Figure 8.6 shows the Latent Dirichlet Allocation Operator in a process. The resulting example sets of this operator contain the topic-distributions, the document-topic distributions and the estimated parameters for the label distribution for sLDA.

8.2.4. Latent Factor Models

Using the Word-Vectors collected into a Term-Document Matrix, we can easily perform LSA via a SVD. The RapidMiner operator **Singular Value Decomposition** extracts the singular values and the singular vectors from an example set. This example set is used as numeric matrix. The operator extracts only the left singular vectors. To extract the right singular vectors, we transpose the data set by the **Transpose** operator and apply the SVD. Given the number of components (factors to be extracted), the operator result is an example set containing the singular vectors and an example set containing the singular values. In Figure 8.14, we illustrate the operator in an example process.

8. Software and Integration

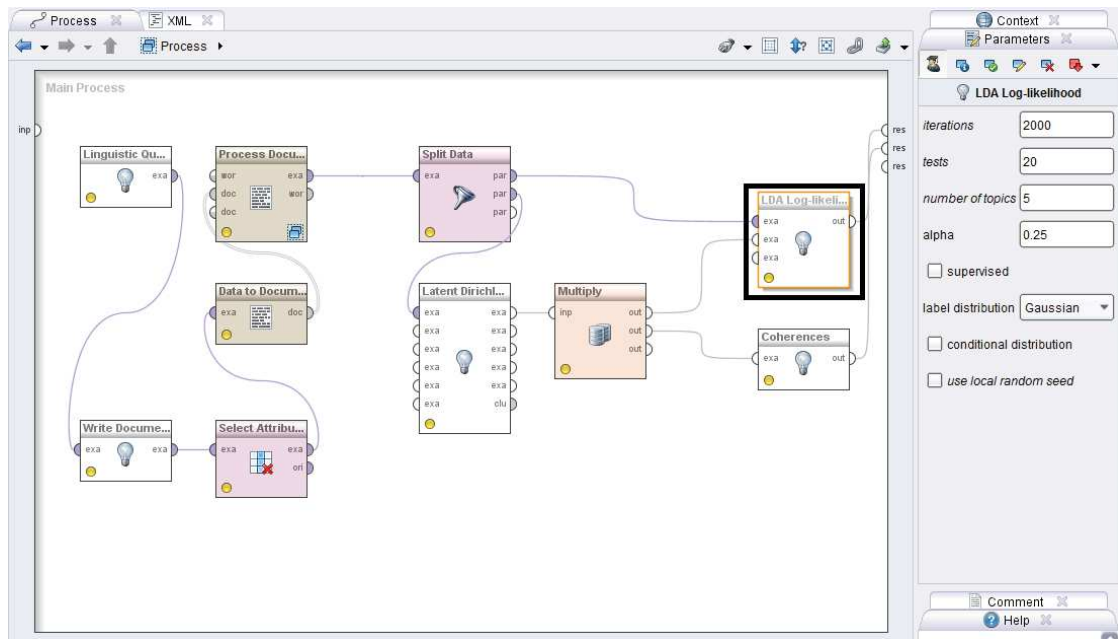


Figure 8.8.: Log-likelihood operator to calculate the likelihood of a test document collection based on sequential Monte Carlo sampling.

8.2.5. Evaluation

Different methods to evaluate latent factor and latent topic models can be used via the **Coherence** operator and the **LDA Log-likelihood** operator, respectively the **LSA Log-likelihood** operator. The Coherence operator takes as input an example set containing word probabilities for topics or factors as a result from the Latent Dirichlet Allocation and the SVD operator. We leverage the Palmetto Toolbox [RBH15] to estimate the different coherence measures based on the top words. From the word probabilities the most likeliest words (the number is given by a parameter) are used for the coherences. To use this operator we need a Lucence-based index from a large text collection that is used as reference. We use the Wikipedia articles to generate such an index that contains coherence values using co-occurrences and relative frequencies. We calculated such indices from German and English using the Palmetto library and the Wikipedia corpora from the Institute of the German Language. The LDA/LSA Log-likelihood Operator need no additional resources. We calculate the likelihoods of a test set of documents by Sequential Monte Carlo methods. As input, the LDA Log-likelihood Operator takes a set of test documents as Word-Vectors with occurrence data in an example set and the word-topic distributions resulting from the LDA operator. The number of iterations specifies the number of Monte Carlo Samples for the estimation of the likelihood. To reduce variance in the likelihood estimation, a number of independent tests are performed. The result is an example set that contains for each test the log-likelihood. The LSA Log-likelihood Operator takes as input an example set with Word-Vectors as test input and the factor representation of the words as second input. This factor representation are for example the left-singular vectors from the Term-Document Matrix extracted by a Singular

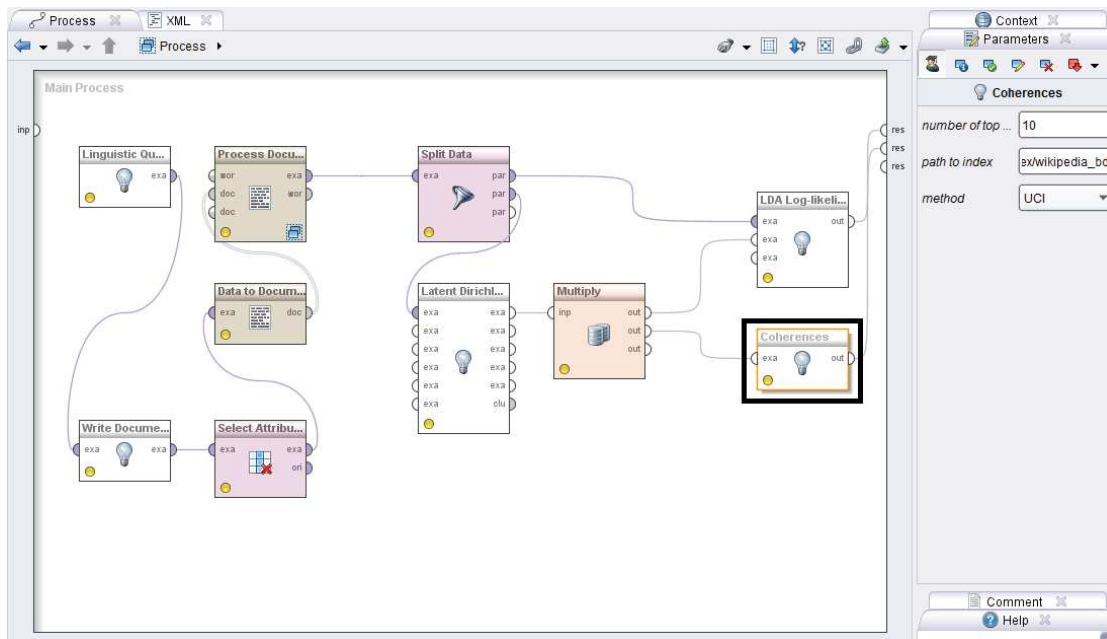


Figure 8.9.: Coherence operator to estimate standard coherence measures using the Palmetto library.

Value Decomposition. The LSA Log-likelihood Operator performs also Sequential Monte Carlo methods but the probabilities are based on distances in the factor representation of the documents and the words. As additional parameters, both operators can estimate joint likelihoods of the documents and possible given labels like time stamps.

8.2.6. Results

The results from the latent factor and latent topic models can be use either in tabular or example set form or in special formats for visualization. Using the results as it is given from the topic models operators, we get two example set containing the topic-word distributions and document-topic distribution. As additional attribute we report the most likeliest topic for each word and each document in the example sets. For factor models using for instance Singular Value Decomposition on the Term-Document Matrix, the factors are given as vectors in an example set and can be used in similar ways as the results from the topic models. In the Figures 8.10 and 8.11, the example sets from the results of LDA are shown as they are internally represented in Rapid-Miner.

Additionally, we implemented an export of the results from the latent variable methods and the corpora for visualization by the DFR-Browser from Andrew Goldstone². To process the documents from the corpus for information extraction needed for the visualization, we implemented the **Write Document Reference** operator. As shown in Figure 8.12, from an example

²<https://github.com/agoldst/dfr-browser>

8. Software and Integration

Row No.	Doc	Topic	Topic_0	Topic_1	Topic_2	Topic_3	Topic_4
1	1	2	0.026939	0.085714	0.695510	0.060408	0.131429
2	2	2	0.164444	0.177778	0.222222	0.217778	0.217778
3	3	1	0.012235	0.925647	0.013647	0.014588	0.033882
4	4	1	0.068293	0.702439	0.033171	0.165854	0.030244
5	5	2	0.037241	0.045517	0.830345	0.051034	0.035862

Figure 8.10.: Document-topic distribution: For each document, one example contains the document number, the distribution over the topics and the most likeliest topic (Topic).

Row No.	Word	Word_id	Topic	Topic_0	Topic_1	Topic_2	Topic_3	Topic_4
1	AT	1	3	0.000021	0.000023	0.000023	0.002602	0.000020
2	Abb	2	4	0.000057	0.000036	0.000035	0.000050	0.007320
3	Abbildung	3	3	0.000053	0.000031	0.000029	0.000610	0.000074
4	Abdruck	4	0	0.001411	0.000079	0.000033	0.000049	0.000214
5	Abdrucke	5	3	0.000175	0.000149	0.000166	0.000204	0.000066

Figure 8.11.: Topic-word distribution: For each word, one example contains the word, the word id, the distribution over the topics and the most likeliest topic (Topic).

set containing texts with information about a title, author, publication date and source as attribute information are saved locally where the visualization tool DFR-Browser finds them. The DFR-Browser can be started as a web server and the visualization can be seen in web browser like Firefox.

8.3. Diachronic Linguistics Process

To perform diachronic linguistic tasks, we use the Latent Dirichlet Allocation operator for sLDA. From the linguistic corpora, we take the information about publication date to extract labels for each document. The attribute **date** from the resulting example set from the TEI or Linguistic Query Operator contains the time information as string. First, we need to convert this into a numerical value by the operators **Nominal to Date** and **Date to Numeric**. Here, the concrete date format (for example "yyyy-MM-dd") must be given and we need to specify the time unit into which we transform the date to numeric (for example years since 1900). This is illustrated on the left in Figure 8.13 (most important operators are framed). After the extraction of the information for visualization by the Write Document References operators, we select the text attribute **text.attr** and the date attribute by the **Select Attribute** operator and process the text to Word-Vectors by the text processing operators. Before the date attribute can be used for

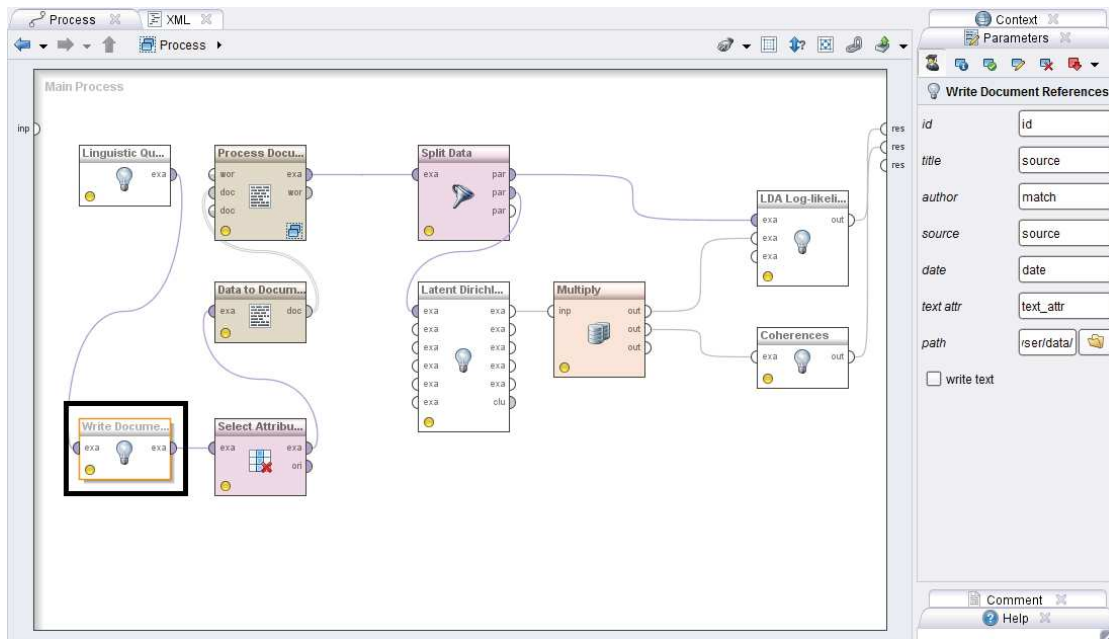


Figure 8.12.: Write Document Reference operator: Writes formatted references from the document collection for visualization by the DFR-Browser.

temporal topic modeling, we need to assign it to the **label** role via the **Set Role** operator. Now, the documents can be used together with the date attribute to extract topics and to estimate label distributions. For temporal topic modeling, we need to check the **supervised** check box in the Latent Dirichlet Allocation operator. We also need to specify with which distribution the labels shall be modeled. For time stamps we can use the Beta, the Uniform and the Gompertz distribution. As results, we get besides the topic-word distributions and the document-topic distributions also the parameters that are estimated by MLE during the topic modeling for the corresponding label distribution.

8.4. Variety Linguistics Process

For variety linguistic tasks to compare and match text collections, we implemented factor models with distribution matching in the **Distribution Matching** operator. Given the Word-Vector representations of two text collection the operator extracts latent factors such that on the subspace spanned by these factors the documents from both collections have a similar distribution. The operator expects two inputs. The first input is a Term-Document Matrix as example set from a text collection with a certain distribution. The second input is a Term-Document Matrix from a second text collection with a different distribution. The results are two example sets containing the projections of the Word-Vectors from the document collections onto the subspace spanned by the factors. In Figure 8.14, an example process for variety linguistic by distribution matching is shown. There are two implementations available. First, a distribution match based

8. Software and Integration

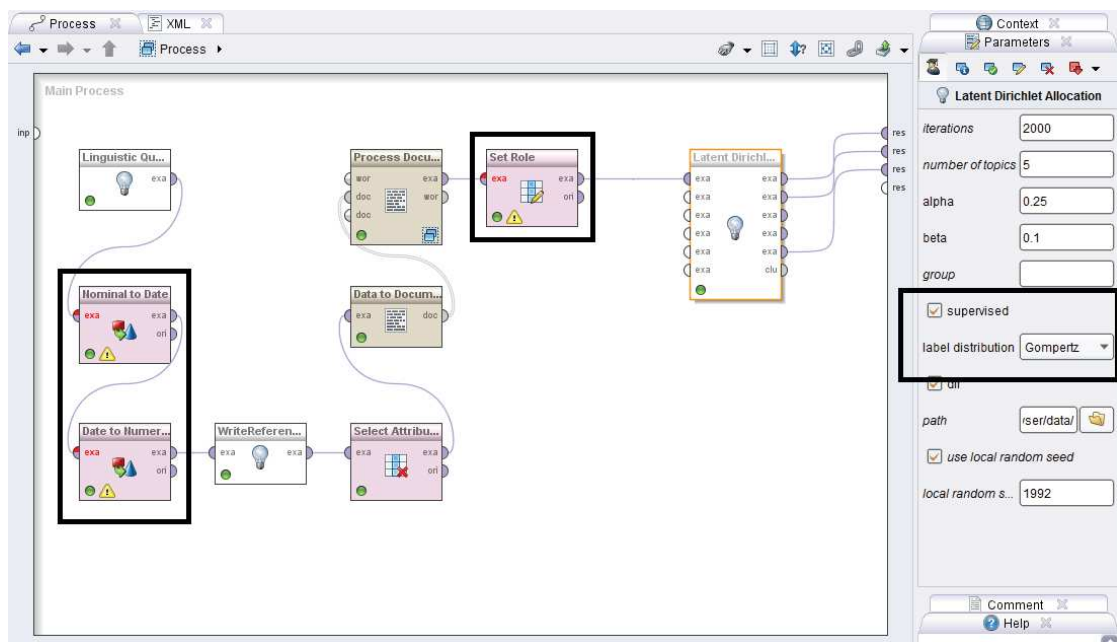


Figure 8.13.: Process for Diachronic Linguistics: From a linguistic data source, we retrieve a KWIC-list with time information. The date is used as numeric label in temporal topic modeling.

on a Singular Value Decomposition extracts factors as the singular vectors of the union of both term-document matrices. Second, we implemented an online method for distribution matching, by efficiently solving an optimization problem through Stochastic Gradient Descent directly on a matrix manifold. We implemented the SGD in Matlab in the ManOpt library [BMAS13] for general Riemann manifolds. To use this method, we need Matlab to be installed and the ManOpt library.

8.5. Software in Application

We successfully integrated our developed methods and the Corpus Linguistics Plugin into the research and teaching of linguistics through the modern language resources from WebLicht and DWDS. Our developed methods proved useful for modern corpus linguistic research.

8.5.1. Integration into WebLicht

WebLicht is a virtual environment for annotating linguistic data. Users load up their document collections or retrieved KWIC-lists from a language resource and perform different annotation steps that enrich their data. So far, WebLicht offered only syntactic annotations like PoS tagging automatically. Here, we integrated latent variable methods into WebLicht as a WebService from RapidMiner and the Corpus Linguistics Plugin into the tool chain of WebLicht services. At the

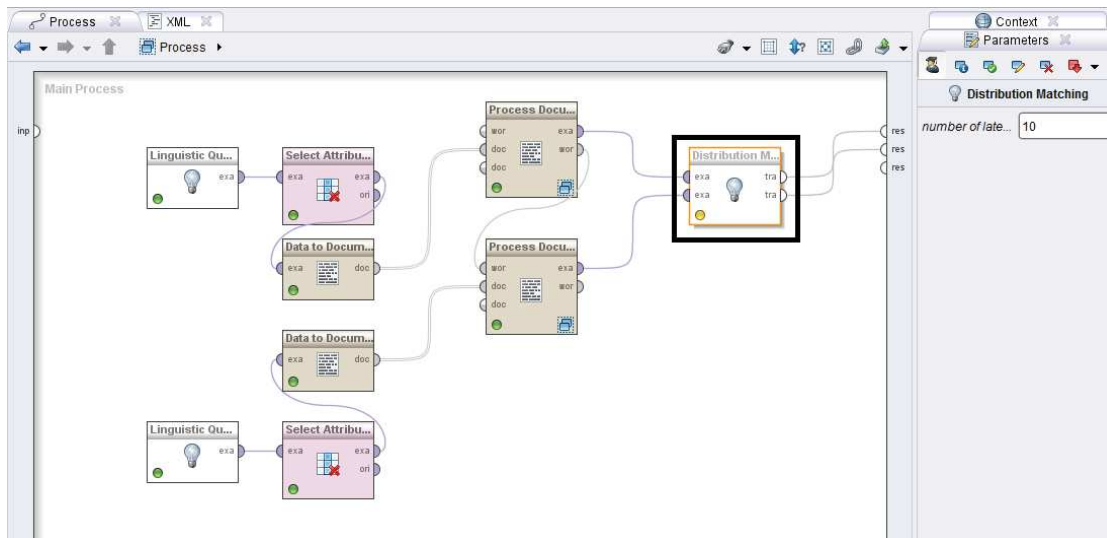


Figure 8.14.: Process for Variety Linguistics.

moment, we successfully deployed Latent Dirichlet Allocation with Gibbs sampling. The results from a tool chain that contains our LDA operator can be visualized by the DFR-Browser.

8.5.2. Integration into DWDS

At the Berlin Brandenburger Academia of Science, “Das Wörterbuch der deutschen Sprache” (DWDS) is developed and maintained. DWDS is a linguistic research environment that can be accessed via a web interface. As described in the introduction, several language resources are accessible. Besides different corpora, dictionaries are especially important for research in lexicography. At the moment we test the Integration of our developed methods into the Dictionary of the German Language. Large corpora can be used to find word senses that are present in a current dictionary. These senses are automatically assigned to examples from KWIC-lists to perform a disambiguation. Besides this, we can use the latent variables to find new senses that are not present in a current dictionary. Currently, we implemented only the assignment of text examples to possible senses from a dictionary.

9. Summary and Conclusion

In this final chapter, we give a resume of the content of the thesis and draw conclusions. After a summary, we conclude with the lesson learned and the impact of this thesis in the research communities of computer science and corpus linguistics as well as the use of the developed software in corpus linguistic teaching. In a final outlook we envision how the developments of this thesis will influence future research and teaching in corpus linguistics, computer science and more.

9.1. Summary

In this thesis, we developed methods and performed extensive studies on latent variable methods for corpus linguistics. Starting with an extensive survey on latent variable methods, the mathematical and geometrical foundations of the thesis are introduced. This survey gives an overview on latent factor models and latent topic models. For the latent factor models, the geometrical interpretation of documents as vectors and factorizations of the Term-Document Matrix are explained and motivated to use for the extraction of latent concepts from digital corpora to extract hidden word senses or subjects in documents. Similarly, latent topic models are introduced as probabilistic model with documents and words as random variables. The hidden concepts in the corpora are modeled as latent random variables. To evaluate the quality of the latent variable methods for the linguistic tasks in diachronic and variety linguistics, qualitative and quantitative methods are explained. Additional methods to evaluate latent variable methods for diachronic linguistics are introduced to complete the collection of quantitative evaluation methods.

For diachronic and variety linguistics in large heterogeneous language resources, regularized versions of the latent variable methods are introduced and motivated. This allows the interpretation of the latent variable methods as optimization over latent variables and for additional regularizations that use further information from the language resources.

In extensive use cases, new efficient latent variable methods are developed to perform diachronic and variety linguistic tasks for large digital corpora in heterogeneous language resources. For diachronic linguistics an attention based topic model as regularized latent variable model is developed. In many experiments, our attention based temporal topic model is compared with standard LDA and a state-of-the-art temporal topic model. For variety linguistics, an efficient method to extract latent factors that are regularized to match different corpora is proposed. Casting the variety linguistic into a Domain Adaptation task, a number of experiments are done to show the benefit qualitatively and quantitatively. Non-linear extensions by kernel methods and efficient approximations are introduced to extract non-linear factors on large digital corpora.

All developed methods are implemented in the Corpus Linguistics Plugin for the software tool RapidMiner. An extensive reference of operators that implement the methods is given together

9. Summary and Conclusion

with examples how the use cases from above are compiled.

9.2. Conclusion and Impact

Large digital corpora from heterogeneous language resources offer valuable information sources for language analysis. The plethora of different resources and the amount of documents from the corpora necessitate the use of automatic methods to extract information from the data to validate and extract linguistic hypotheses. This thesis investigated Natural Language Processing methods for research and teaching on corpus linguistics. In lexicography and semantics, latent variable methods were developed to perform diachronic and variety linguistic tasks. Hidden concepts from large corpora are automatically extracted and associated with word senses and subjects in documents. The impact of the results are shown in use cases and by the application of the methods in modern language resource infrastructures like WebLicht. Now, we can efficiently perform diachronic and variety linguistic tasks on large digital corpora to support linguistic research in lexicography and semantics.

To finally conclude this thesis, we describe the impact in the research communities from corpus linguistics and computer science. We explain the significance of the works presented in this thesis and give an outlook of how these works can influence the future.

9.2.1. Impact

We measure the impact of this thesis in terms of relevance to computer science and corpus linguistics. Next, we give detailed arguments for the significance in these fields.

Significance in Corpus Linguistics

In several talks at different conventions, parts from this thesis were successfully presented as corpus linguistic research. At the *4th General Virtual Competency meeting of DARIAH-EU* in Rome [BP14], the joint work in Computer Mediated Communication analysis with the help of the methods developed in this thesis was presented. In Berlin at the *Digital Humanities Summit 2015* [BPS15], a poster highlighted our empirical work with language resources. At the *Digital Humanities im deutschsprachigen Raum* convention 2015 [PM15], the methods from this thesis were introduced to the Digital Humanities community. Further, at the *Forum CA3* from Clarin-D in Hamburg [Pöl16], the methods and software from this thesis were demonstrated.

Additionally, in papers at conferences from the Digital Humanities, parts of this thesis have been published. At the *Clarin 2014* conference and the workshop *Language Technology for Cultural Heritage, Social Sciences, and Humanities 2014*, the motivation and some parts of the pre-studies of this thesis have been published [PB14b, PB14a]. In additional technical reports, the developed methods and parts of the use cases are published [PBB14] as part of the BmBF project KobRA.

The Corpus Linguistics Plugin from this thesis is already used for teaching corpus linguistics. Starting with the first joint seminar *Korpusgestützte Analyse internetbasierter Kommunikation*

mit Hilfe von *Data-Mining* in 2014, the Artificial Intelligence Group and the Institute of German Language and Literature introduced the Corpus Linguistics Plugin and the RapidMiner for teaching linguistic courses. The results of this seminar were presented at the Konvens conference in [BLMP14] and at the convention *Neue Wege in der Nutzung von Korpora: Data-Mining für die textorientierten Geisteswissenschaften*, Berlin-Brandenburgische Akademie der Wissenschaften [Mor15]. Further, at Mannheim University, the plugin is used in a *Projekt Seminar* in the Master studies *Spache und Kommunikation* and a dedicated seminar *Korpusbasierte Sprachanalyse*.

Significance in Computer Science

In computer science, the impact of this these can be evaluated by the publications at international conference for Data Mining and Machine Learning. The main use cases are published on peer reviewed conferences. At the conferences *Text, Speech and Dialog 2015* [PBMS15] and at the *Urban Data Mining Workshop at the International Conference of Machine Learning 2015* [Pöl15b] some parts of the diachronic linguistic use case are published. The use case in variety linguistics for regularized linear factor models is to be published as journal paper in the Springer *Machine Learning Journal*. Parts of the use case for regularized non-linear factor models is published at the *International Conference on Pattern Recognition Applications and Methods 2015* [Pöl15a] and at the *First International Workshop on Learning over Multiple Contexts* [Pöl14] at the ECML 2014.

Besides the publications on international conferences or journals in computer science, this thesis provides significant contributions to computer science. The evaluation methods from Chapter 3 that measure the quantitative quality of latent variable methods that include temporal information provides a valuable contribution for evaluating topic and factor models. In computer science such evaluation methods play a big role to measure the quality of existing and new methods. The attention based temporal topic model from Chapter 5 provides new points of view to combine Bayesian approaches with Diffusion Models. This allows for meaningful models that comply to real world processes like attentions. The efficient optimization method on matrix manifolds from Chapter 7 offers new solutions to extract latent factors from large document collections. Especially from the efficiency point of view, this method provides an online solution that can be used on big data scenarios. Previous approaches on the other hand use closed form solutions that result in prohibitively large memory consumptions.

9.2.2. Outlook

Looking into the crystal ball, we learn several potential applications and links to future research and developments in corpus linguistics and computer science.

Future Use in Corpus Linguistics

In the future, we will see that the methods developed in this thesis become a helpful resource in corpus linguistic research and teaching. Linguistic research already uses the software from this thesis to perform large linguistic studies. The usage of automatic analysis methods grows

9. Summary and Conclusion

rapidly in linguistic research due to the amounts of natural language data that become available. The methods and the software from this thesis become a part of these developments. The same can be said for linguistic teaching. In order to educate linguists to perform automatic analyses to extract and validate linguistic hypothesis in large digital corpora or general document collections, the methods and the software from this thesis support teaching in corpus or computer linguistic courses.

Future Use in Computer Science

In the field of computer science, understanding natural language is becoming more and more important. Consequently, the methods developed in this thesis are very useful in computer science in the future. Furthermore, the methods and the studies from this thesis will be further developed in the future. The attentional topic model for instance is combined with different temporal distributions that model not only growth and decline, but also periodicity. Further, the optimization on matrix manifolds to extract latent factors is the starting point to general regularized latent factor models.

Also in the future, there are several applications for the methods from this thesis. Large IT companies have already started to invest into analyzing the language of their customers and users. The contributions from this thesis help to understand user interactions with modern IT services from Google and Apple. As modern IT companies turn to Human Computer Interactions (HCI) and chat bots to interact with users, the methods from this thesis are used to create language models of the users. Temporal aspects like attentions to certain subjects and varieties in language play an important role to create UCI systems. On large corpora of user data, latent variable methods as those from this thesis are used to extract user preferences. Further, in big social media companies like Facebook, the user content is already used to analysis their behavior. With the growth of social media content, the contributions of this thesis help to investigate the variation or the temporal distribution in the language of the users.

A. Appendix

A.1. Collaborations

Some parts of this thesis emerged from collaborations with colleagues and researchers from the Artificial Intelligence Group and the Institute of German Language and Literature at TU Dortmund University. Next, the chapters that contain joint work are listed and concrete collaborations are described. All chapters that are not listed, stem from pure work alone from the author of this thesis.

A.1.1. Chapter 1

The pre-studies in the introduction chapter results from collaborations with the Institute of German Language and Literature at TU Dortmund University in the context of the BmBF project KobRA¹. The pre-studies are additionally published together with the co-researchers from the project, Thomas Bartz, Prof. Angelika Storrer and Prof. Katharina Morik.

A.1.2. Chapter 5

The attention based temporal topic model for the use case in diachronic linguistics was jointly worked out by Prof. Kristian Kersting, Elena Erdmann and the author of this thesis. The initial idea of the attention model is from Prof. Christian Baukhage, Prof. Kristian Kersting and Dr. Fabian Hadiji from the Bonn University. In a joint paper, this collaborations is protocoled. The content in this chapter is the contribution of the author of the thesis. Additionally, the corpus of the Spiegel articles used in some experiments originates from Prof. Hendrick Müller from the Institute of Journalistic at TU Dortmund University.

A.1.3. Chapter 7

The methods from the use case of variety linguistics stem purely from the author of this thesis. The same is true for the analysis and the evaluation of the method. A journal publication about this method was a joint publication with Dr. Wouter Duivesteijn and Prof. Katharina Morik.

A.2. Publications

Finally, we list all publications that have been produced for this thesis. We report papers that already have been accepted at conferences or journals. Additionally, we report the papers that are at the moment under review of major conferences or journals.

¹<http://www.kobra.tu-dortmund.de>

A.2.1. Accepted Papers

1. Christian Pölitz, Thomas Bartz, Katharina Morik, and Angelika Störrer. Investigation of word senses over time using linguistic corpora. In *Text, Speech, and Dialogue - 18th International Conference, TSD'15*, pages 191–198, Cham, CH, 2015. Springer
2. Christian Pölitz. Modelling time and location in topic models. In *Proceedings of the 2nd International Workshop on Mining Urban Data co-located with 32nd International Conference on Machine Learning*, volume 1392 of *MUD'15*, pages 95–96, online, 2015. CEUR Workshop Proceedings
3. Christian Pölitz. Distance based active learning for domain adaptation. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, volume 1 of *ICPRAM '15*, pages 296–303, Setubal, PT, 2015. scitepress
4. Lothar Lemnitzer, Christian Pölitz, Jörg Didakowski, and Alexander Geyken. Combining a rule-based approach and machine learning in a good-example extraction task for the purpose of lexicographic work on contemporary standard german. In *Proceedings of the eLex 2015 conference*, eLex '15, pages 21–31, Ljubljana, SL. Trojina, Institute for Applied Slovene Studies / Lexical Computing Ltd
5. Alexander Geyken, Christian Pölitz, and Thomas Bartz. Using a maximum entropy classifier to link good corpus examples to dictionary senses. In *Proceedings of the eLex 2015 conference*, eLex '15, pages 304–314, Ljubljana, SL. Trojina, Institute for Applied Slovene Studies / Lexical Computing Ltd
6. Christian Pölitz. Subset based hilbert space projections for transfer learning. First International Workshop on Learning over Multiple Contexts, LMCE 2014, 2014
7. Christian Pölitz and Thomas Bartz. Enhancing the possibilities of corpus-based investigations: Word sense disambiguation on query results of large text corpora. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanitie*, LaTeCH EACL'14, pages 42–46, Stroudsburg, PA, USA, 2014. ACL
8. Thomas Bartz, Christian Pölitz, Katharina Morik, and Angelika Störrer. Using data mining and the clarin infrastructure to extend corpus-based linguistic research. *Jan Odiijk (Ed.): Selected Papers from the CLARIN 2014 Conference*, pages 1–13, 2014
9. Michael Beißwenger, Harald Lungen, Eliza Margaretha, and Christian Pölitz. Mining corpora of computer-mediated communication. In Gertrud Faaß and Josef Ruppenhofer, editors, *Proceedings of the 12th edition of the KONVENS conference*, volume 1 of *Analysis of linguistic features in Wikipedia talk pages using machine learning methods*, pages 42 – 47, Hildesheim, DE, 2014. University of Hildesheim
10. Thomas Bartz, Nadja Radtke, and Christian Pölitz. Digitale korpora in der internetlexikographie. *Lexicographica*, 30:605–610, 2014

11. Christian Pölitz and Thomas Bartz. Using data mining and the clarin infrastructure to extend corpus-based linguistic research. The CLARIN Annual Conference, 2014

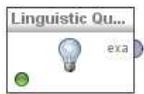
A.2.2. Papers under Review

Additional, there are papers under review from this thesis. The use case of diachronic linguistics with attentional topic models is submitted to the EMNLP. The use case of variety linguistics with regularized factor models via Stochastic Gradient Descent on matrix manifolds is to be published in the *Machine Learning Journal*. The use case for corpus linguistics with non-standard corpora by sparsity inducing priors on LDA is submitted to the DMNLP Workshop at ECML 2016. Finally, the last part of the use case on variety linguistics by regularized non-linear factor models is submitted to the KDML 2016.

A.3. Operator Reference

Next, we summarize the operators for the RapidMiner Plugin that implement the methods from this thesis. For clarity, we group the references into subsections.

A.3.1. Data Imports



Linguistic Query Operator : Interface to a linguistic corpora via a data base server as maintained by Berlin Brandenburg Academia of Science

Parameter	Description
query	The linguistic query for a digital corpus.
encapsulation	Character that indicates position of query match in the results.
sample size	Number of snippets in the results KWIC-list.
encoding	Character encoding (UTF-8)
data source	Corpus to query.
context size	Number of sentences before and after the sentence that contains the query match.
extract lemmas	Retrieve additional lemma information for each word (if available).
extract tags	Retrieve additional Parts-of-Speech for each word (if available).
Output	Example set containing KWIC-list.

A. Appendix



TEI Query Operator: Stream reader for TEI formatted XML-files

Parameter	Description
query	The query as regular expression on the TEI-file.
file	Path to the TEI-file to be queried.
context	Environment in which we look for query match (posting, paragraph or sentence level).
context size	Number of characters before and after the regular expression match.
sample size	Number of snippets in the results KWIC-list.
regular expression	Query input mask regular expression on the TEI-file (with editor).
encoding	Character encoding (UTF-8)
output	Example set containing KWIC-list.



WordNet Operator: Query word relations from word net files

Parameter	Description
query	The word for the WordNet relations.
word net resource	Path to the word net data base files.
output	Example set containing word net relations.



Word Profiles Operator: Query word profiles from DWDS server

Parameter	Description
query	The word for the word profiles relations.
number of results	The number of related words by word profiles.
output	Example set containing word profiles relations.

A.3.2. Latent Topic Models



Latent Dirichlet Allocation Operator: Extracts latent topics via LDA.

Parameter	Description
iterations	The number of iterations for the Gibbs sampler.
number of topics	The number of latent topics to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
beta	Meta parameter for Dirichlet prior of topic-word distribution.
group	Group attribute in data set.
supervised	Supervised LDA or unsupervised.
label distribution	The distribution of the document labels (Gauss, Beta, Uniform, Gompertz).
dfr	Print results in format for DFR-Browser.
path	Path to save files form DFR-Browser.
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of documents as Word-Vectors with occurrences. (for supervised an additional attribute with label role must be available.)
output 1	Example set containing topic-word distributions.
output 2	Example set containing document-topic distributions.
output 3	Example set containing number of assignments of topics to words (for evaluation).
output 4	Example set containing number of assignments of topics to any word (for evaluation).
output 5	Example set containing parameters of the estimated label distributions.

A. Appendix



Hierarchical Latent Dirichlet Allocation

Operator: Extracts latent topics via LDA including hierarchies between the topics.

Parameter	Description
iterations	The number of iterations for the Gibbs sampler.
number of topics	The number of latent topics to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
beta	Meta parameter for Dirichlet prior of topic-word distribution.
group	Group attribute in data set.
supervised	Supervised LDA or unsupervised.
label distribution	The distribution of the document labels (Gauss, Beta, Uniform, Gompertz).
dfr	Print results in format for DFR-Browser.
path	Path to save files form DFR-Browser.
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of documents as Word-Vectors with occurrences. (for supervised an additional attribute with label role must be available.)
output 1	Example set containing topic-word distributions.
output 2	Example set containing document-topic distributions.
output 3	Example set containing number of assignments of topics to words (for evaluation).
output 4	Example set containing number of assignments of topics to any word (for evaluation).
output 5	Example set containing parameters of the estimated label distributions.



Latent Dirichlet Allocation with Word Features

Operator: Extracts latent topics via LDA and includes word features and relations via priors.

Parameter	Description
iterations	The number of iterations for the Gibbs sampler.
number of topics	The number of latent topics to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
beta	Meta parameter for Dirichlet prior of topic-word distribution.
lambda	
gamma	
number of word groups	for group lasso based prior.
a	Meta parameter for group lasso penalty.
prior	used prior
dfr	Print results in format for DFR-Browser.
path	Path to save files form DFR-Browser.
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of documents as Word-Vectors with occurrences. (for supervised an additional attribute with label role must be available.)
input 2	Example set containing word relations.
output 1	Example set containing topic-word distributions.
output 2	Example set containing document-topic distributions.
output 3	Example set containing number of assignments of topics to words (for evaluation).
output 4	Example set containing number of assignments of topics to any word (for evaluation).
output 5	Example set containing parameters of the estimated label distributions.

A. Appendix



Dirichlet Multinomial Regression Operator:

Extracts latent topics via LDA.

Parameter	Description
iterations	The number of iterations for the Gibbs sampler.
number of topics	The number of latent topics to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
beta	Meta parameter for Dirichlet prior of topic-word distribution.
lambda	
sigma	
group	Group attribute in data set.
dfr	Print results in format for DFR-Browser.
path	Path to save files form DFR-Browser.
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of documents as Word-Vectors with occurrences.
input 2	Additional document attributes.
output 1	Example set containing topic-word distributions.
output 2	Example set containing document-topic distributions.
output 3	Example set containing number of assignments of topics to words (for evaluation).
output 4	Example set containing number of assignments of topics to any word (for evaluation).
output 5	Example set containing parameters of the estimated label distributions.

A.3.3. Latent Factor Models



Distribution Matching: Extracts latent factors that match distributions.

Parameter	Description
number of factors	The number latent factors to be extracted.
input 1	Example set of documents as Word-Vectors from a certain distribution.
input 2	Example set of documents as Word-Vectors from another distribution.
output 1	Example set of the documents from input 1 projected into the latent factor presentation.
output 2	Example set of the documents from input 2 projected into the latent factor presentation.

A.3.4. Evaluation Methods



LDA Log-likelihood Operator: Estimates log-likelihood on a test collection of sequences of words in document given the results of LDA.

Parameter	Description
iterations	The number of iterations for the Gibbs sampler.
tests	The number of random tests.
number of topics	The number to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
supervised	Supervised LDA or unsupervised.
label distribution	The distribution of the document labels (Gauss, Beta, Uniform, Gompertz).
conditional distribution	
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of test documents as BoW with occurrence data
input 2	Example set of word-topic distribution from an LDA result.
output 1	Example set containing log-likelihoods from each test.

A. Appendix



LSA Log-likelihood Operator: Estimates log-likelihood on a test collection of Word-Vectors by distance based distribution estimation.

Parameter	Description
iterations	The number of iterations for the Gibbs sampler.
tests	The number of random tests.
number of topics	The number to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
smoothing gamma	Supervised LSA (PLS) or unsupervised.
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of test word-vectors.
input 2	Example set of words in factor representation.
output 1	Example set containing log-likelihoods from each test.



Coherence Operator: Estimates standard coherence values based on top ranked word in each topic.

Parameter	Description
number of top words	The number of top ranked words in each topic used to estimate coherence values.
path to index	The path to the lucene index files for Palmetto.
method	Used coherence measure. (UCI,UMass,NPMI)
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
input 1	Example set of topic-word distributions from an LDA results.
output 1	Example set containing the coherence value for each topic.

Nomenclature

α, β	Dirichlet meta parameter
B	The Beta function
C	A corpus or a document collection
$\text{Dir}(\alpha)$	Dirichlet distribution
d	A single document
\mathbf{d}	The sequence of words in document d
E	Matrix containing the singular values of $X \in \mathbb{R}^{M,V}$
e	A singular value
K_x	A kernel matrix for documents mapped into an RKHS
\hat{K}	Low dimensional kernel matrix approximation in kPCA
L	Matrix containing the left singular vectors of $X, \in \mathbb{R}^{M,M}$
\mathbf{l}_i	The loadings vectors in PLS
λ, γ, ϕ	Variational parameters
\mathbf{l}	A left singular vector
M	Number of documents
$\text{Mult}(\phi)$	Multinomial distribution
\mathbf{m}	Metaparameters for prior distributions and temporal distributions
N	Number of tokens (in a single document)
n_{t_i}	The number of times topic t_i has been assigned to any token
n_d	The number of times document d has been assigned to any topic
$n_{t_i,w}$	The number of times topic t_i has been assigned to word w
n_{d,t_i}	The number of times document d has been assigned to topic t_i
P	Projection matrix
\hat{p}	Empirical distribution of documents in a corpus
$p(w)$	Word probability
$p(w t), \beta_t$	Topic-word distribution
$p(t d), \theta_d$	Document-topic distribution
R	Matrix containing the right singular vectors of $X, \in \mathbb{R}^{V,V}$
\mathbf{r}	A right singular vector
Θ	Parameters for any latent factor method
T	Number of factors or topics
t	A single factor or topic
\mathbf{t}	A sequence of topic assignments: $\mathbf{t} = (t_1, \dots, t_N)$
t_i	A single topic assignment from a sequence \mathbf{t}
\mathbf{t}^{-i}	A sequence of topic assignments without assignment i : $\mathbf{t}^{-i} = (t_1, \dots, t_{i-1}, t_{i+1} \dots t_n)$
\mathbf{t}_{+t}^{-i}	A sequence of topic assignments with t_i replaced by t : $\mathbf{t}_{+t}^{-i} = (t_1, \dots, t_{i-1}, t, t_{i+1} \dots t_n)$
V	Number of words the in vocabulary
\mathbb{V}	The vocabulary of a corpus
\mathbf{v}^i	i_{th} latent factors as vector in the VSM

A.3. Operator Reference

\mathbf{v}'	The transpose of any vector \mathbf{v}
W, H	Positive matrices factorizing X in NNMF
w_i	Word i in vocabulary \mathbb{V}
w_n	n th word token in sequence (w_1, \dots, w_N)
\mathbf{w}_d	Word-Vector $\in \mathbb{R}^V$
\mathbf{w}^i	i th basis vector in VSM associated with word w_i
X	Term-Document Matrix, $\in \mathbb{R}^{M,V}$
X'	The transpose of the Term-Document Matrix, $\in \mathbb{R}^{V,M}$

Glossary

- Accuracy** Quality of a classification as frequency of correct classified examples. 156, 162
- Bag-of-Words** Representation of documents by the set of contained words. 24, 41, 78, 145, 205
- Comma Separated Value** Tabular data as string with fields separated by commas. 205
- Das Wörterbuch der deutschen Sprache** A collection of digital corpora, dictionaries and statistics of written German language. 181, 205
- Dialing and DWDS Concordance** Open source (LGPL) search engine developed specially to meet the needs of linguistic researchers. 171, 205
- Dirichlet Multinomial Regression** Modelling the meta parameter of a Dirichlet distribution as a regression. 84, 205
- Document-Vector** Representation of words as vector in a Euclidean space. 34, 89
- Domain Adaptation** Methods to transfer information across differently distributed distributions. 138, 139, 145–149, 153–157, 159, 160, 162–167, 183, 205
- JavaScript Object Notation** Data format for fast exchanging purpose. 205
- Kernel Partial Least Squares** Extracts non-linear factors that maximally align with given document labels. 43, 205
- Kernel Principal Component Analysis** Extracts non-linear factors with Principal Component Analysis on a kernel matrix. 42, 145, 205
- Key-Word-in-Context** A usage example for a word of interest with its context. 12, 205
- Language Model** Probabilistic model of the generation of word sequences in documents. 25, 26, 122, 205
- Latent Dirichlet Allocation** Models and extract topics as latent random variables. Additional the document-topic and the topic-term distribution have a Dirichlet prior. 48, 205
- Latent Semantic Analysis** Extracts linear factors from term-document matrix. 34, 205

- Maximum Likelihood Estimation** Parameter estimation that maximize the likelihood given data. 25, 101, 128, 205
- Maximum Mean Discrepancy** Distance measure between distributions as supremum norm in an RKHS of the expectation functional of the distributions. 140, 206
- Multinomial Model** Representation of documents as sequence of words drawn from Multinomial distributions. 25, 32, 205
- Non-negative Matrix Factorization** Extracts non-negative factors from term-document matrix. 39, 206
- Normalized Mutual Information** Measure of how likely two words are associated with a normalized measure. 73, 206
- Partial Least Squares** Extracts linear factors that maximally align with given document labels. 38, 206
- Pointwise Mutual Information** Measure of how likely two words are associated. 72, 206
- Principal Component Analysis** Extracts linear factors as orthonormal bases for a matrix. 42, 206
- Probabilistic Latent Semantic Analysis** Models and extract topics as latent random variables from word occurrences in documents. 45, 48, 206
- Reproducing Kernel Hilbert Space** A Hilbert space that allows for point evaluations via inner products. 41, 206
- Singular Value Decomposition** Matrix factorization. 34, 180, 206
- Stiefel manifold** The set $M(p, q) = \{P \mid P \in \mathbb{R}^{q \times p}, P^T \cdot P = I\}$, together with an inner product \cdot . 142–144, 149, 150, 156, 157, 160–163, 165
- Stochastic Gradient Descent** Optimization. 53, 141, 180, 189, 206
- supervised LDA** Extension of LDA that jointly models document labels, words and topic. 85, 96, 206
- Support Vector Machine** Supervised learning method for classification and regression based on large margins in a vector space. 61, 157, 206
- Term Frequency** Frequency of a word in a document. The number of occurrences of the word token divided by the number of all word tokens in the document. 24, 206
- Term Frequency Inverse Document Frequency** Normalized frequency of a word in a document. The term frequency of a word multiplied by a normalization term that is small for words that appear in many other documents. 24, 206

Term-Document Matrix Matrix containing the Word-Vector from a corpus as column vectors. 25, 32, 34, 35, 38, 39, 42, 71, 78, 82, 86, 175, 177, 179, 183, 199

Text Encoding Initiative Electronic text format for certain linguistic research. 170, 206

tokenizer Separation of document or texts as string into word tokens. 173

Topics over Time Topic model that additionally models time as Beta distributed random variable. 96, 101, 107, 205, 206

Vector Space Model Representation of documents as vectors in a Euclidean space. 23, 32, 34, 206

Word-Vector Representation of documents as vector in a Euclidean space. 24–30, 33–36, 38, 39, 41, 43, 46, 66, 67, 70, 71, 78, 79, 81, 82, 87–91, 136–142, 148–154, 156, 158, 165, 172–175, 177, 179, 199, 203

WordNet Lexical data base of English words. In so called Synsets, words are described that are synonym. Relations between Synsets build a graph and describe relation and similarities between words. 83, 90, 91, 125

Acronyms

- @TM** Attentional Topic Model. 101, 102, 107–109, 111–116, *Glossary*: Attentional Topic Model
- BoW** Bag-of-Words. 24, 41, 141, 145, *Glossary*: Bag-of-Words
- CSV** Comma Separated Value. 170, *Glossary*: Comma Separated Value
- DA** Domain Adaptation. *Glossary*: Domain Adaptation
- DDC** Dialing and DWDS Concordance. 171, *Glossary*: Dialing and DWDS Concordance
- DMR** Dirichlet Multinomial Regression. 84, 86, 125, *Glossary*: Dirichlet Multinomial Regression
- DWDS** Das Wörterbuch der deutschen Sprache. 109, 171, 181, *Glossary*: Das Wörterbuch der deutschen Sprache
- JSON** JavaScript Object Notation. 171, 172, *Glossary*: JavaScript Object Notation
- kPCA** Kernel Principal Component Analysis. 42, 43, 78, 82, 83, 198, *Glossary*: Kernel Principal Component Analysis
- kPLS** kernel Partial Least Squares. 43, 59, 78, *Glossary*: Kernel Partial Least Squares
- KWIC** Key-Word-in-Context. 12, 14, 16, 107, 171, 172, 181, *Glossary*: Key-Word-in-Context
- LDA** Latent Dirichlet Allocation. 48, 50, 51, 53, 55, 58–64, 71, 76, 83, 85, 86, 95, 96, 98, 99, 101–104, 107–109, 112–116, 122–126, 128, 129, 131–133, 170, 173, 177, 202, 206, *Glossary*: Latent Dirichlet Allocation
- LM** Language Model. 25–27, *Glossary*: Language Model
- LSA** Latent Semantic Analysis. 34–39, 41, 42, 44, 46, 64, 71, 78, 83, 86, 87, 89, 170, 175, *Glossary*: Latent Semantic Analysis
- MLE** Maximum Likelihood Estimation. 25, 123, 179, *Glossary*: Maximum Likelihood Estimation
- MM** Multinomial Model. 25–30, 33, 34, 44, 81, *Glossary*: Multinomial Model

Acronyms

- MMD** Maximum Mean Discrepancy. 140–142, 144, 146–148, 158, 159, 163, 165, *Glossary*: Maximum Mean Discrepancy
- NNMF** Non-negative Matrix Factorization. 39, 41, 46, 47, 78, 82, 83, 86, 199, *Glossary*: Non-negative Matrix Factorization
- NPMI** Normalized Pointwise Mututal Information. 73, *Glossary*: Normalized Mututal Information
- PCA** Principal Component Analysis. 42, *Glossary*: Principal Component Analysis
- PLS** Partial Least Squares. 38, 39, 41, 43, 59, 78, 79, 198, *Glossary*: Partial Least Squares
- pLSA** probabilistic LSA. 45, 46, 59, 64, 82, 83, *Glossary*: Probabilistic Latent Semantic Analysis
- PMI** Pointwise Mututal Information. 72, 73, *Glossary*: Pointwise Mututal Information
- RKHS** Reproducing Kernel Hilbert Space. 41–43, 78, 82, 140, 142, 198, 202, *Glossary*: Reproducing Kernel Hilbert Space
- SGD** Stochastic Gradient Descent. 53, 141–143, 149, 150, 156–161, 165, 180, *Glossary*: Stochastic Gradient Descent
- sLDA** supervised LDA. 64, 85, 86, 96–98, 101, 102, 175, 178, *Glossary*: supervised LDA
- SVD** Singular Value Decomposition. 34, 36–38, 42, 71, 170, 175, *Glossary*: Singular Value Decomposition
- SVM** Support Vector Machine. 156, 157, 170, *Glossary*: Support Vector Machine
- TEI** Text Encoding Initiative. 170, 172, 173, 178, *Glossary*: Text Encoding Initiative
- TF** Term Frequency. 24, 173, *Glossary*: Term Frequency
- TF-IDF** Term Frequency Inverse Document Frequency. 24, 152, 173, *Glossary*: Term Frequency Inverse Document Frequency
- TOT** Topics over Time. 96, 98, 102, 108, 109, 112–116, *Glossary*: Topics over Time
- VSM** Vector Space Model. 23–30, 33, 34, 67, 70, 81, 87, 90, 136, 138, 141, 198, 199, *Glossary*: Vector Space Model

Bibliography

- [AG07] Galen Andrew and Jianfeng Gao. Scalable training of ℓ_1 -regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, volume 22 of *ICML '07*, pages 33–40, New York, NY, USA, 2007. ACM.
- [AMS08] Pierre-Antoine Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2008.
- [AS13] Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics*, volume 10 of *IWCS '13*, pages 13–22, New York, NY, USA, 2013. ACM.
- [AWST09] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press.
- [AX12] Amr Ahmed and Eric Xing. Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream. *Computing Research Repository*, abs/1203.3463, 2012.
- [AZC09] David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, volume 24 of *ICML '09*, pages 25–32, New York, NY, USA, 2009. ACM.
- [AZCR11] David Andrzejewski, Xiaojin Zhu, Mark Craven, and Benjamin Recht. A framework for incorporating general domain knowledge into latent dirichlet allocation using first-order logic. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, volume 22 of *IJCAI '11*, pages 1171–1177, Palo Alto, CA, USA, 2011. AAAI Press.
- [Bas69] Frank Bass. A new product growth for model consumer durables. *Management Science*, 15(5):215–227, 1969.
- [BBC⁺10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning Journal*, 79(1-2):151–175, 2010.

Bibliography

- [BBS09] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10:2137–2155, December 2009.
- [BDBCP07] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In Bernhard Schölkopf, John Platt, and Thomas Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19 of *NIPS '06*, pages 137–144, Cambridge, MA, US, 2007. MIT Press.
- [BDP07] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, volume 45 of *ACL '07*, pages 440–447, Stroudsburg, PA, USA, 2007. ACL.
- [BEG⁺12] Michael Beißwenger, Maria Ermakova, Alexander Geyken, Lothar Lemnitzer, and Angelika Storrer. A TEI schema for the representation of computer-mediated communication. *Journal of the Text Encoding Initiative [Online]*, (3), 2012.
- [BGBZ07] Jordan L. Boyd-Graber, David M. Blei, and Xiaojin Zhu. A topic model for word sense disambiguation. In *Empirical Methods in Natural Language Processing, EMNLP-CoNLL '07*, pages 1024–1033, Stroudsburg, PA, USA, 2007. ACL.
- [BGJ10] David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):7:1–7:30, February 2010.
- [BGJT04] David Blei, Thomas Griffiths, Michael Jordan, and Joashua Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. volume 16 of *NIPS '03*, pages 17–24, Cambridge, MA, USA, 2004. MIT Press.
- [BHK15] Christian Bauckhage, Fabian Hadiji, and Kristian Kersting. How viral are viral videos? In *Proceedings of the Ninth International Conference on Web and Social Media*, volume 9 of *ICWSM '15*, pages 22–30, Palo Alto, CA, USA, 2015. AAAI Press.
- [BHLS13] Mahsa Baktashmotlagh, Mehrtash Harandi, Brian Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the 2013 International Conference on Computer Vision*, volume 14 of *ICCV '13*, New York City, NY, USA, 2013. IEEE Press.
- [BJM11] Francis Bach, Rodolphe Jenatton, and Julien Mairal. *Optimization with Sparsity-Inducing Penalties (Foundations and Trends(R) in Machine Learning)*. Now Publishers Inc., Hanover, MA, USA, 2011.
- [BK14] Christian Bauckhage and Kristian Kersting. Strong regularities in growth and decline of popularity of social media services. *Computing Research Repository*, abs/1406.6529, 2014.

- [BL06a] David Blei and John Lafferty. Correlated topic models. *Advances in Neural Information Processing Systems*, 19:147, 2006.
- [BL06b] David Blei and John Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, volume 21 of *ICML '06*, pages 113–120, New York, NY, USA, 2006. ACM.
- [BLMP14] Michael Beißwenger, Harald Lungen, Eliza Margaretha, and Christian Pölit. Mining corpora of computer-mediated communication. In Gertrud Faaß and Josef Ruppenhofer, editors, *Proceedings of the 12th edition of the KONVENS conference*, volume 1 of *Analysis of linguistic features in Wikipedia talk pages using machine learning methods*, pages 42 – 47, Hildesheim, DE, 2014. University of Hildesheim.
- [BMAS13] Nicolas Boumal, Bamdev Mishra, Pierre-Antoine Absil, and Rodolphe Sepulchre. Manopt: a Matlab toolbox for optimization on manifolds. *arXiv preprint arXiv:1308.5200 [cs.MS]*, 2013.
- [BNJ03] David Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.
- [BNR10] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. *Computing Research Repository*, abs/1006.4046, 2010.
- [Bon13] Silvere Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.
- [Bot98] Léon Bottou. Online algorithms and stochastic approximations. In *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK, 1998.
- [BP14] Michael Beißwenger and Christian Pölit. Analyzing CMC corpora using machine learning methods: Report from the KobRA project. 4th General Virtual Competency Centre (VCC) meeting of DARIAH-EU, September 2014.
- [BPMS14] Thomas Bartz, Christian Pölit, Katharina Morik, and Angelika Storrer. Using data mining and the clarin infrastructure to extend corpus-based linguistic research. *Jan Odijk (Ed.): Selected Papers from the CLARIN 2014 Conference*, pages 1–13, 2014.
- [BPS15] Thomas Bartz, Christian Pölit, and Angelika Storrer. Erprobung innovativer data-mining-verfahren für die empirische arbeit mit strukturierten sprachressourcen. Posterpräsentation auf dem Digital Humanities Summit, March 2015.
- [BRP14] Thomas Bartz, Nadja Radtke, and Christian Pölit. Digitale korpora in der internetlexikographie. *Lexicographica*, 30:605–610, 2014.

Bibliography

- [BS06] Elena Bonelli and John Sinclair. Corpora. In Keith Brown, editor, *Encyclopedia of Language and Linguistics (Second Edition)*, pages 206 – 220. Elsevier, Oxford, UK, second edition edition, 2006.
- [BWS04] Gökhan H. Bakr, Jason Weston, and Bernhard Schölkopf. Learning to find pre-images. In *Advances in Neural Information Processing Systems*, volume 16 of *NIPS '03*, pages 449–456, New York, NY, USA, 2004. MIT Press.
- [Byn77] Theodora Bynon. *Historical Linguistics*:. Cambridge University Press, Cambridge, USA, MA, 009 1977.
- [CJ98] Noah Coccaro and Daniel Jurafsky. Towards better integration of semantic predictors in statistical language modeling. In *ICSLP*. ISCA, 1998.
- [CLTW09] Bo Chen, Wai Lam, Ivor Tsang, and Tak-Lam Wong. Extracting discriminative concepts for domain adaptation in text mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 15 of *KDD '09*, pages 179–188, New York, NY, USA, 2009. ACM.
- [CML⁺13] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Discovering coherent topics using general knowledge. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, volume 22 of *CIKM '13*, pages 209–218, New York, NY, USA, 2013. ACM.
- [CMP03] Xavier Carreras, Lluís Màrquez, and Lluís Padró. A simple named entity extractor using adaboost. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, volume 4 of *CONLL '03*, pages 152–155, Stroudsburg, PA, USA, 2003. ACL.
- [Cru06] Alexander Cruden. *A complete concordance to the Holy Scriptures of the Old and New Testaments*. Printed and sold by Kimber, Conrad & Co, Philadelphia, PA, USA, 1806.
- [CSF⁺12] Rita Chattopadhyay, Qian Sun, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Multisource domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data (TKDD) - Special Issue on the Best of SIGKDD 2011*, 6(4):18:1–18:26, 2012.
- [CSG09] Kevin R. Canini, Lei Shi, and Thomas L. Griffiths. Online inference of topics with latent dirichlet allocation. In David V. Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *AISTATS-09*, pages 65–72, Cambridge, MA, USA, 2009. Journal of Machine Learning Research - Proceedings Track.
- [Cul65] Karel Culík. Machine translation and connectedness between phrases. In *First International Conference on Computational Linguistics*, volume 1 of *COLING '65*, Stroudsburg, PA, USA, 1965. ACL.

- [CWS12] Yutian Chen, Max Welling, and Alexander Smola. Super-samples from kernel herding. *Computing Research Repository*, abs/1203.3472, 2012.
- [Dav10] Mark Davies. The corpus of contemporary American english as the first reliable monitor corpus of english. *Literary and Linguistic Computing*, pages 447–464, 2010.
- [DDF⁺90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
- [DE09] Gabriel Doyle and Charles Elkan. Accounting for burstiness in topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, volume 24 of *ICML '09*, pages 281–288, New York, NY, USA, 2009. ACM.
- [DG92] Alessandro Duranti and Charles Goodwin. Rethinking context: An introduction. In Alessandro Duranti and Charles Goodwin, editors, *Rethinking Context: Language as an Interactive Phenomenon*, chapter 1, page 142. Cambridge University Press, Cambridge, MA, USA, 1992.
- [DG13] Jörg Didakowski and Alexander Geyken. From dwds corpora to a german word profile methodological problems and solutions. In *In Network Strategies, Access Structures and Automatic Extraction of Lexicographical Information*, pages 43–52, Mannheim, GE, 2013. Mannheim: Institut für Deutsche Sprache. (OPAL - Online publizierte Arbeiten zur Linguistik X/2012).
- [DHWX13] Avinava Dubey, Ahmed Hefny, Sinead Williamson, and Eric Xing. A nonparametric mixture model for topic modeling over time. In *SIAM International Conference on Data Mining*, volume 13 of *SDM '13*, pages 530–538, Philadelphia, PA, USA, 2013. SIAM.
- [DLP08] Chris Ding, Tao Li, and Wei Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics and Data Analysis*, 52(8):3913–3927, 2008.
- [DLR77] Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
- [DSP06] Miroslav Dudík, Robert E. Schapire, and Steven J. Phillips. Correcting sample selection bias in maximum entropy density estimation. In *Advances in Neural Information Processing Systems*, volume 18 of *NIPS '05*, Cambridge, MA, USA, 2006. MIT Press.
- [EAS99] Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, April 1999.

Bibliography

- [Fir57] John Firth. A synopsis of linguistic theory 1930-55. 1952-59:1–32, 1957.
- [FK79] Nelson Francis and Henry Kucera. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979.
- [GBR⁺08] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel method for the two-sample problem. *Computing Research Repository*, abs/0805.2368, 2008.
- [Gey07] Alexander Geyken. The dwds corpus: A reference corpus for the german language of the 20th century. In *Fellbaum, Christiane (Hg.): Collocations and Idioms: Linguistic, lexicographic, and computational aspects*, pages 23–41, London, UK, 2007. Continuum International Publishing Group.
- [GG84] Stuart Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741, 1984.
- [GG06] Zoubin Ghahramani and Thomas L. Griffiths. Infinite latent feature models and the indian buffet process. In Y. Weiss, B. Schölkopf, and J.C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18 of *NIPS '05*, pages 475–482. MIT Press, Cambridge, MA, USA, 2006.
- [GGS13] Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *ICML '13*, pages 222–230, New York, NY, USA, 2013. ACM.
- [GJG⁺15] Samah Gad, Waqas Javed, Sohaib Ghani, Niklas Elmqvist, Thomas Ewing, Keith Hampton, and Naren Ramakrishnan. Themedelta: Dynamic segmentations over temporal topic models. *IEEE Transactions on Visualization and Computer Graphics*, 21(5):672–685, 2015.
- [GK03] Mark Girolami and Ata Kabán. On an equivalence between plsi and lda. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 433–434, New York, NY, USA, 2003. ACM.
- [GLC11] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of the 2011 International Conference on Computer Vision*, volume 13 of *ICCV '11*, pages 999–1006, New York City, NY, USA, 2011. IEEE Press.
- [GPB] Alexander Geyken, Christian Pölit, and Thomas Bartz. Using a maximum entropy classifier to link good corpus examples to dictionary senses. In *Proceedings of the eLex 2015 conference*, eLex '15, pages 304–314, Ljubljana, SL. Trojina, Institute for Applied Slovene Studies / Lexical Computing Ltd.

- [Gra12] Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 24 of *CVPR '12*, pages 2066–2073, Washington, DC, USA, 2012. IEEE Computer Society.
- [Grö73] Bernhard Gröschel. Linguistische Grundbegriffe und Methodenüberblick. *Lingua*, 32(1):165 – 172, 1973.
- [GS90] Alan E. Gelfand and Adrian F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990.
- [GS04] Thomas Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.
- [GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.
- [Har81] Zellig S. Harris. *Papers on Syntax*, chapter Distributional Structure, pages 3–22. Springer Netherlands, Dordrecht, NL, 1981.
- [HBB10] Matthew Hoffman, David Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 23 of *NIPS '10*, pages 856–864, Red Hook, NY, USA, 2010. Curran Associates, Inc.
- [He12] Yulan He. Incorporating sentiment prior knowledge for weakly supervised sentiment analysis. 11(2):4:1–4:19, June 2012.
- [HF97] Birgit Hamp and Helmut Feldweg. Germanet - a lexical-semantic net for german. In *In Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15, 1997.
- [HJM08] David Hall, Daniel Jurafsky, and Christopher Manning. Studying the history of ideas using topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, volume 12 of *EMNLP '08*, pages 363–371, Stroudsburg, PA, USA, 2008. ACL.
- [HK13] Markus Hofmann and Ralf Klinkenberg. *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC, London, UK, 2013.
- [Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, pages 13–30, 1963.
- [Hof99] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

Bibliography

- [Hof00] Knut Hofland. A self-expanding corpus based on newspapers on the web. In *2nd Edition of its Language Resources and Evaluation Conference*, LREC '00, Paris, FR, 2000. European Language Resources Association.
- [Hot33] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 1933.
- [HPJ99] Thomas Hofmann, Jan Puzicha, and Michael Jordan. Learning from dyadic data. In *Advances in Neural Information Processing Systems*, volume 11 of *NIPS '99*, pages 466–472. Curran Associates, Inc., Red Hook, NY, USA, 1999.
- [HSG⁺07a] Jiayuan Huang, Alexander Smola, Arthur Gretton, Karsten Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, volume 19 of *NIPS '06*, pages 601–608, Red Hook, NY, USA, 2007. Curran Associates, Inc.
- [HSG⁺07b] Jiayuan Huang, Alexander Smola, Arthur Gretton, Karsten Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In Bernhard Schölkopf, John Platt, and Thomas Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19 of *NIPS '06*, pages 601–608, Cambridge, MA, USA, 2007. MIT Press.
- [Hud96] Richard Hudson. *Sociolinguistics*. Cambridge Press, Cambridge, UK, 1996.
- [Hun06] Susan Hunston. Corpus linguistics. In Keith Brown, editor, *Encyclopedia of Language and Linguistics (Second Edition)*, pages 234 – 248. Elsevier, Amsterdam, NL, second edition edition, 2006.
- [Jär94] Timo Järvinen. Annotating 200 million words: The bank of english project. In *Proceedings of the 15th Conference on Computational Linguistics*, volume 1 of *COLING '94*, pages 565–568, Stroudsburg, PA, USA, 1994. ACL.
- [Jen06] Johan Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906.
- [JGJS99] Michael Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, November 1999.
- [Kai58] Henry Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, 1958.
- [KG10] Wolfgang Klein and Alexander Geyken. Das Digitale Wörterbuch der Deutschen Sprache (DWDS). Volume 26:79–96, December 2010.
- [KK09] Marc Kupietz and Holger Keibel. The Mannheim German Reference Corpus (dereko) as a basis for empirical linguistic research. In *Working Papers in Corpus-based Linguistics and Language Education*, pages 53–59, Tokyo, JP, 2009. Tokyo University of Foreign Studies (TUFS).

- [KMT12] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the nyström method. *Journal of Machine Learning Research*, 13:981–1006, April 2012.
- [Kol79] Matthew Koll. Weird: An approach to concept-based information retrieval. *SIGIR Forum*, 13(4):32–50, April 1979.
- [KSO87] Maurice Kendall, Alan. Stuart, and Keith Ord, editors. *Kendall’s Advanced Theory of Statistics*. Oxford University Press, Inc., New York, NY, USA, 1987.
- [KW92] Jacek Kuczyski and Henryk Woniakowski. Estimating the largest eigenvalue by the power and lanczos algorithms with a random start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, 1992.
- [Lan95] Ken Lang. NewsWeeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, volume 10 of *ICML ’95*, pages 331–339, San Mateo, CA, USA, 1995. Morgan Kaufmann publishers Inc.
- [LD97] Thomas Landauer and Susan Dumais. A solution to plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211–240, 1997.
- [Lin07] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Comput.*, 19(10):2756–2779, October 2007.
- [LJSJ09] Simon Lacoste-Julien, Fei Sha, and Michael Jordan. Disclda: Discriminative learning for dimensionality reduction and classification. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, volume 21 of *NIPS ’08*, pages 897–904. Curran Associates, Inc., Red Hook, NY, USA, 2009.
- [LM06] Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, volume 21 of *ICML ’06*, pages 577–584, New York, NY, USA, 2006. ACM.
- [LN89] Dong Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(3):503–528, 1989.
- [LPDG] Lothar Lemnitzer, Christian Pölitz, Jörg Didakowski, and Alexander Geyken. Combining a rule-based approach and machine learning in a good-example extraction task for the purpose of lexicographic work on contemporary standard german. In *Proceedings of the eLex 2015 conference*, eLex ’15, pages 21–31, Ljubljana, SL. Trojina, Institute for Applied Slovene Studies / Lexical Computing Ltd.
- [LPSS⁺14] David Lopez-Paz, Suvrit Sra, Alex Smola, Zoubin Ghahramani, and Bernhard Schölkopf. Randomized nonlinear component analysis. In Eric P. Xing and

Bibliography

- Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 29 of *ICML '14*, pages 1359–1367, New York, NY, USA, 2014. ACM.
- [LS01] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13 of *NIPS '00*, pages 556–562, Red Hook, NY, USA, 2001. Curran Associates, Inc.
- [LW10] Haifeng Liu and Zhaohui Wu. Non-negative matrix factorization with constraints. In Maria Fox and David Poole, editors, *Association for the Advancement of Artificial Intelligence*, *AAAI '10*, Palo Alto, CA, USA, 2010. AAAI Press.
- [LWD⁺13] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 25 of *ICCV '13*, pages 2200–2207, New York City, NY, USA, 2013. IEEE Press.
- [LYRL04] David Lewis, Yiming Yang, Tony Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machin Learning Research*, 5:361–397, December 2004.
- [MB88] Geoffrey McLachlan and Kay Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.
- [MB08] Jon Mcauliffe and David Blei. Supervised topic models. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20 of *NIPS '07*, pages 121–128. Curran Associates, Inc., Red Hook, NY, USA, 2008.
- [MBS13] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. *Computing Research Repository*, abs/1301.2115, 2013.
- [Mer09] James Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London*, 209:415–446, 1909.
- [Mil95] George Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, November 1995.
- [ML02] Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, *UAI'02*, pages 352–359, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
- [ML14] Eliza Margaretha and Harald Lüngen. Building linguistic corpora from wikipedia articles and discussions. *Journal of Language Technology and Computational Linguistics*, 29(2):59–82, 2014.

- [MM12] David M. Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. *CoRR*, abs/1206.3278, 2012.
- [Mor15] Katharina Morik. Rapidminer als werkzeug fr die textorientierten geisteswissenschaften. *Neue Wege in der Nutzung von Korpora: Data-Mining fr die textorientierten Geisteswissenschaften*, Oktober 2015.
- [MS99] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.
- [MWT⁺11] David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, volume 11 of *EMNLP '11*, pages 262–272, Stroudsburg, PA, USA, 2011. ACL.
- [Myu03] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90–100, February 2003.
- [Nav09] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1–10:69, February 2009.
- [NBB11a] David Newman, Edwin Bonilla, and Wray Buntine. Improving topic coherence with regularized topic models. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, volume 24 of *NIPS '10*, pages 496–504. Curran Associates, Inc., Red Hook, NY, USA, 2011.
- [NBB11b] David Newman, Edwin V. Bonilla, and Wray Buntine. Improving topic coherence with regularized topic models. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24 of *NIPS '11*, pages 496–504, Red Hook, NY, USA, 2011. Curran Associates, Inc.
- [NCL07] Ramesh Nallapati, William Cohen, and John Lafferty. Parallelized variational em for latent dirichlet allocation: An experimental evaluation of speed and scalability. In *Seventh IEEE International Conference on Data Mining Workshops*, volume 7 of *ICDMW '07*, pages 349–354, Red Hook, NY, USA, 2007. Curran Associates, Inc.
- [NJW01] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, volume 14 of *NIPS '01*, pages 849–856, Cambridge, MA, USA, 2001. MIT Press.
- [NLGB10] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 100–108, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Bibliography

- [NLS14] Christian Andersson Naesseth, Fredrik Lindsten, and Thomas Schön. Sequential monte carlo for graphical models. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26 of *NIPS '13*, pages 1862–1870, Red Hook, NY, USA, 2014. Curran Associates, Inc.
- [Nor12] Matthew North. *Data mining for the masses*. Global Text Project, 2012.
- [NQC13] Jie Ni, Qiang Qiu, and Rama Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, volume 13 of *CVPR '13*, pages 692–699, New York City, NY, USA, 2013. IEEE Press.
- [NSS11] Nasir Naveed, Sergej Sizov, and Steffen Staab. Att: Analyzing temporal dynamics of topics and authors in social media. In *Proceedings of the 3rd International Web Science Conference*, volume 3 of *WebSci '11*, pages 1:1–1:7, New York, NY, USA, 2011. ACM.
- [NSWA08] David Newman, Padhraic Smyth, Max Welling, and Arthur Asuncion. Distributed inference for latent dirichlet allocation. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20 of *NIPS '07*, pages 1081–1088. Curran Associates, Inc., Red Hook, NY, USA, 2008.
- [Par95] Barbara Partee. Lexical semantics and compositionality. *An invitation to cognitive science: Language*, 1:311–360, 1995.
- [PB14a] Christian Pölitz and Thomas Bartz. Enhancing the possibilities of corpus-based investigations: Word sense disambiguation on query results of large text corpora. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanitie*, LaTeCH EACL'14, pages 42–46, Stroudsburg, PA, USA, 2014. ACL.
- [PB14b] Christian Pölitz and Thomas Bartz. Using data mining and the clarin infrastructure to extend corpus-based linguistic research. The CLARIN Annual Conference, 2014.
- [PBB14] Christian Pölitz, Thomasa Bartz, and Michael Beißwenger. Überwachte und unüberwachte disambiguierung von kwic-snippets bei der suche in groen textkorpora. data-mining-verfahren des kobra-projekts. Technical report, TU Dortmund University, Computer Science Department, 2014.
- [PBMS15] Christian Pölitz, Thomas Bartz, Katharina Morik, and Angelika Störrer. Investigation of word senses over time using linguistic corpora. In *Text, Speech, and Dialogue - 18th International Conference*, TSD'15, pages 191–198, Cham, CH, 2015. Springer.

- [PC98] Jay Ponte and Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, volume 21 of *SIGIR '98*, pages 275–281, New York, NY, USA, 1998. ACM.
- [PD65] Eugene Pendergraft and Nell Dale. Automatic linguistic classification. In *First International Conference on Computational Linguistics*, volume 1 of *COLING '65*, Stroudsburg, PA, USA, 1965. ACL.
- [Pin94] Steven Pinker. *The language instinct: How the mind creates language*. William Morrow and Company, New York City, NY, USA, 1994.
- [PL08] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, January 2008.
- [PM15] Christian Pölitz and Katharina Morik. Big data und data mining in den digital humanities. Von Daten zu Erkenntnissen: Digitale Geisteswissenschaften als Mittler zwischen Information und Interpretation, Februar 2015.
- [Pö114] Christian Pölitz. Subset based hilbert space projections for transfer learning. First International Workshop on Learning over Multiple Contexts, LMCE 2014, 2014.
- [Pö115a] Christian Pölitz. Distance based active learning for domain adaptation. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, volume 1 of *ICPRAM '15*, pages 296–303, Setubal, PT, 2015. scitepress.
- [Pö115b] Christian Pölitz. Modelling time and location in topic models. In *Proceedings of the 2nd International Workshop on Mining Urban Data co-located with 32nd International Conference on Machine Learning*, volume 1392 of *MUD'15*, pages 95–96, online, 2015. CEUR Workshop Proceedings.
- [Pö116] Christian Pölitz. Data mining for large scale corpus linguistic. Forum CA3 2016, June 2016.
- [PPM04] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet::similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, volume 2 of *HLT-NAACL–Demonstrations '04*, pages 38–41, Stroudsburg, PA, USA, 2004. ACL.
- [PSC⁺10] James Petterson, Alexander Smola, Tibrio Caetano, Wray Buntine, and Shraavan Narayanamurthy. Word features for latent dirichlet allocation. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23 of *NIPS '10*, pages 1921–1929, Red Hook, NY, USA, 2010. Curran Associates, Inc.
- [PTKY09] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. In *Proceedings of the 21st International*

Bibliography

- Joint Conference on Artificial Intelligence*, volume 21 of *IJCAI'09*, pages 1187–1192, New York City, NY, USA, 2009. IEEE Press.
- [PTKY11] Sinno Jialin Pan, Ivor Tsang, James Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, Feb 2011.
- [PTRV98] Christos Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, volume 7 of *PODS '98*, pages 159–168, New York, NY, USA, 1998. ACM.
- [PZS⁺13] Shimei Pan, Michelle X. Zhou, Yangqiu Song, Weihong Qian, Fei Wang, and Shixia Liu. Optimizing temporal topic segmentation for intelligent text visualization. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 339–350, New York, NY, USA, 2013. ACM.
- [QPS09] Novi Quadrianto, James Petterson, and Alexander Smola. Distribution matching for transduction. In *Advances in Neural Information Processing Systems*, volume 22 of *NIPS '08*, pages 1500–1508, New York City, NY, USA, 2009. MIT Press.
- [RBH15] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eight International Conference on Web Search and Data Mining*, Shanghai, February 2-6, 2015.
- [RG65] Herbert Rubenstein and John Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, October 1965.
- [RHNM09] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
- [RR08] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20 of *NIPS '07*, pages 1177–1184, Red Hook, NY, USA, 2008. Curran Associates, Inc.
- [RT02] Roman Rosipal and Leonard Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 2:97–123, March 2002.
- [SB88] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

- [SB13] James Scott and Jason Baldridge. A recursive estimate for the predictive likelihood in a topic model. In *Artificial Intelligence and Statistics Conference*, volume 31 of *JMLR Proceedings '13*, pages 527–535, Cambridge, MA, USA, 2013. MIT Press.
- [SCGF12] Ming Shao, Carlos Castillo, Zhenghong Gu, and Yun Fu. Low-rank transfer subspace learning. *IEEE International Conference on Data Mining*, 12:1104–1109, 2012.
- [SN10] Issei Sato and Hiroshi Nakagawa. Topic models with power-law using pitman-yor process. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 673–682, New York, NY, USA, 2010. ACM.
- [SNK⁺08a] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems*, volume 20 of *NIPS '07*, Red Hook, NY, USA, 2008. Curran Associates, Inc.
- [SNK⁺08b] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems*, volume 20 of *NIPS '07*, Red Hook, NY, USA, 2008. Curran Associates, Inc.
- [Sok03] Alexey Sokirko. Ddc - a search engine for linguistically annotated corpora. In *Proceedings of Dialogue (published in Russian)*, pages 25–64, Moskow, RU, 2003. Nauka.
- [SR05] Nathan Srebro and Sam Roweis. Time-varying topic models using dependent dirichlet processes. Technical report, 2005.
- [SSM99] Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. Advances in kernel methods. chapter Kernel Principal Component Analysis, pages 327–352. MIT Press, Cambridge, MA, USA, 1999.
- [ST69] Roger Shank and Larry Tesler. A conceptual dependency parser for natural language. In *Third International Conference on Computational Linguistics*, COLING '69, Stroudsburg, PA, USA, 1969. ACL.
- [STC04] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.
- [STG10] Si Si, Dacheng Tao, and Bo Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(7):929–942, 2010.

Bibliography

- [TNW07] Yee Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In B. Schölkopf, J.C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19 of *NIPS '06*, pages 1353–1360. MIT Press, Cambridge, MA, USA, 2007.
- [Tur50] Alan Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.
- [Vap95] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
- [vdMH08] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.
- [WBH12] Chong Wang, David Blei, and David Heckerman. Continuous Time Dynamic Topic Models. *The Computing Research Repository*, abs/1206.3298, 2012.
- [Wel09] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, volume 24 of *ICML '09*, pages 1121–1128, New York, NY, USA, 2009. ACM.
- [Wit53] Ludwig Wittgenstein. *Philosophical Investigations*. Basil Blackwell, Oxford, UK, 1953. (trans. by G.E.M. Anscombe).
- [WM06] Xuerui Wang and Andrew McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 12 of *KDD '06*, pages 424–433, New York, NY, USA, 2006. ACM.
- [WMM09] Hanna Wallach, David Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *Advances in Neural Information Processing Systems*, volume 22 of *NIPS '09*, pages 1973–1981, Red Hook, NY, USA, 2009. Curran Associates, Inc.
- [WMSM09] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, volume 24 of *ICML '09*, pages 1105–1112, New York, NY, USA, 2009. ACM.
- [WR12] Daniel Walker and Eric Ringger. Topics over nonparametric time: A supervised topic model using bayesian nonparametric density estimation. In *Proceedings of the Ninth UAI Bayesian Modeling Applications Workshop*, volume 962 of *BMAW '12*, pages 74–83, online, 2012. CEUR Workshop Proceedings.
- [WXK10] Mirwaes Wahabzada, Zhao Xu, and Kristian Kersting. Topic models conditioned on relations. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III, ECML PKDD'10*, pages 402–417, Berlin, Heidelberg, 2010. Springer-Verlag.

- [WY13] Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.
- [YMM09] Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 937–946, New York, NY, USA, 2009. ACM.
- [YZX12] Jing Yuan, Yu Zheng, and Xing Xie. Discovering regions of different functions in a city using human mobility and POIs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 186–194, New York, NY, USA, 2012. ACM.
- [ZAX09] Jun Zhu, Amr Ahmed, and Eric Xing. Medlda: Maximum margin supervised topic models for regression and classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, volume 24 of *ICML '09*, pages 1257–1264, New York, NY, USA, 2009. ACM.
- [ZCL11] Jia Zeng, William K. Cheung, and Jiming Liu. Learning topic models by belief propagation. *Computing Research Repository*, abs/1109.3437, 2011.
- [ZZW⁺13] Kai Zhang, Vincent Zheng, Qiaojun Wang, James Kwok, Qiang Yang, and Ivan Marsic. Covariate shift in Hilbert space: A solution via surrogate kernels. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *ICML '13*, pages 388–395, New York, NY, USA, 2013. ACM.