On the Use of Language Models and Topic Models in the Web

New Algorithms for Filtering, Classification, Ranking, and Recommendation

Von der Fakultät für Elektrotechnik und Informatik der
Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades eines
DOKTORS DER NATURWISSENSCHAFTEN
Dr. rer. nat.

genehmigte Dissertation

von
Dipl.-Inform. Ralf Krestel
geboren am 11.09.1980
in Kehl am Rhein

2012

Leibniz
Universität
Hannover

Forschungszentrum · Research Center L3S

Referent: Prof. Dr. Wolfgang Nejdl
Koreferent: Prof. Dr. Gerd Stumme

Tag der Promotion: 20.02.2012

Ralf Krestel

On the Use of Language Models and Topic Models in the Web

Abstract

The huge amount of information available in the Web poses various challenges to Web science researchers. Finding interesting and relevant pieces of information is like finding a needle in a haystack. Since most searchable information is either written in or described by natural language, efficient and effective methods to represent natural language are crucial. In this thesis we look into topic models and language models as a mean to structure, compress, and represent textual data. In particular, we look into latent Dirichlet allocation, a generative topic model, and its use for various task scenarios in the Web. We discuss advantages and disadvantages in comparison with language models in the context of common Web applications from areas such as information retrieval, recommender systems, and machine learning.

In particular, we investigate the use of topic models and language models for *filtering*, *classification*, *ranking*, and *recommendation*. These are popular methods to cope with the information overload the average user has to deal with in the Web today. Many applications make use of these methods, such as Web search engines, Web 2.0 platforms, or recommender systems. We present different approaches based on topic model and language model representations for the individual tasks.

Regarding the *filtering* of information, we propose an approach based on support vector machines to predict the importance of news articles. We compare the use of topic models and language models as the underlying representation of the newspaper articles to accurately predict important news automatically. Sentiment *classification* deals with automatically detecting the polarity of texts. This is often done by looking-up sentiment scores for each term in a document in a lexicon. We propose a context-dependent sentiment lexicon based on latent topics identified by latent Dirichlet allocation. The most common task in the Web is probably the *ranking* of information. The huge success of search engines from Google or Yahoo! is based on sophisticated ranking algorithms. We look into diversification of rankings to cover the information needs of as many users as possible. Therefore we not only investigate diversification of Web search results, but also diversification of product review rankings. The *recommendation* of products, friends, news, or events is very popular in e-commerce sites like Amazon, or social Web applications like Facebook. We propose an approach for tag recommendation within folksonomies. We show that combining topic models to overcome the cold start problem with language models to personalize the recommendations is very successful.

Topic Models, Language Models, Latent Dirichlet Allocation, Information Retrieval, Sentiment Analysis, Recommender Systems

Zusammenfassung

Die riesige Menge an verfügbaren Informationen im World Wide Web stellt eine große Herausforderung für Wissenschaftler unterschiedlicher Disziplinen dar. Das Finden von interessanten und relevanten Informationen gleicht dem Finden einer Stecknadel im Heuhaufen. Da die meisten durchsuchbaren Informationen entweder in natürlicher Sprache vorliegen oder mit deren Hilfe beschrieben werden (z.B. in Form von Metadaten), kommt der effizienten und effektiven Representation von Textdaten eine besondere Rolle zu. In dieser Arbeit betrachten wir Topic-Modelle und Language-Modelle, um diese Daten zu strukturieren, zu komprimieren und zu speichern. Insbesondere betrachten wir Latent Dirichlet Allocation, ein generatives Topic-Modell, und dessen Verwendbarkeit innerhalb verschiedener Anwendungsszenarien im Kontext alltäglicher Web Anwendungen aus den Bereichen Information Retrieval, Recommender Systeme, und maschinellem Lernen.

Im einzelnen werden wir den Nutzen von Topic-Modellen und Language Modellen für das *Filtern*, *Klassifizieren*, *Ranken* und *Empfehlen* von Informationen untersuchen. Dies sind populäre Methoden, um mit der Informationsflut im Web fertig zu werden, der der durchschnittliche Benutzer täglich ausgesetzt ist. Viele Anwedungen im Web wie Suchmaschinen, Web 2.0 Platformen oder Recommender Systeme benutzen solche Methoden. In dieser Arbeit werden verschiedene Ansätze vorgestellt welche auf einer Topic- oder Language-Modell Repräsentation basieren und diese Methoden einsetzen.

Für das *Filtern* von Informationen wird ein Ansatz vorgestellt, welcher auf Support Vector Maschinen basierend, die Wichtigkeit von Zeitungsartikeln vorherzusehen versucht. Wir analysieren dafür sowohl Topic-Modelle als auch Language-Modelle als grundlegende Repräsentation der Artikel. Die *Klassifizierung*, basierend auf Sentimentanalyse, versucht die Polarität eines Textes automatisch zu bestimmen. Dafür werden häufig sogenannte Sentimentlexica benutzt. Wir stellen einen Ansatz für ein kontextabhängiges Sentimentlexicon vor, welches auf latenten Themen aufbaut, die durch Latent Dirichlet Allocation automatisch identifiziert werden. Die meist verbreitetste Aufgabe im Web ist wahrscheinlich das *Ranken* von Informationen. Der große Erfolg der Suchmaschinen von Google oder Yahoo! basiert auf ausgeklügelten Ranking-Algorithmen. Wir betrachten das Diversifizieren von Rankings, um so viele Benutzer wie möglich zufrieden zu stellen. Um unsere Ergebnisse zu validieren, diversifizieren wir nicht nur Suchmaschinenergebnisse, sondern auch Produktrezensionen. Das *Empfehlen* von Produkten, Freunden, Nachrichten oder Ereignissen ist sehr populär in E-Commerce Portalen wie Amazon oder bei sozialen Web Anwendungen wie Facebook. In dieser Arbeit stellen wir einen Ansatz zur automatischen Empfehlung von Schlagwörtern innerhalb einer Folksonomy vor. Wir zeigen, dass das Kombinieren von Topic- und Language-Modellen dafür zu sehr guten Ergebnissen führt.

Topic-Modelle, Language-Modelle, Latent Dirichlet Allocation, Information Retrieval, Sentimentanalyse, Recommendersysteme

CONTENTS

LIST OF ABBREVIATIONS

AUC	Area Under the Curve
BEP	Break-Even Point
BOW	Bag-of-Words
CDSL	Context Dependent Sentiment Lexicon
CDSV	Context Dependent Sentiment Value
DAG	Directed Acyclic Graph
IR	Information Retrieval
KLD	Kullback-Leibler Divergence
LDA	Latent Dirichlet Allocation
LM	Language Model
LSA	Latent Semantic Analysis
ML	Machine Learning
NER	Named Entity Recognition
NLP	Natural Language Processing
PAM	Pachinko Allocation Model
PLSA	Probabilistic Latent Semantic Analysis
PMI	Pointwise Mutual Information
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
TFV	Term Frequency Vector
TM	Topic Model
VSM	Vector Space Model

CHAPTER
ONE

INTRODUCTION

1.1 Problem Description and Proposed Solutions

The Web is the largest information storage medium on earth and keeps growing. In 2011, more than 200 million top-level domain names were registered [89]. With this huge amount of information, automatic methods to process data and assist users in retrieving relevant information are indispensable. *Filtering* information, *classifying* data, retrieving and *ranking* relevant information, and *recommending* items in a personalized or unpersonalized way provide support for users to cope with the huge amount of data in the Web.

Although video data accounts for most traffic in the Internet [88], textual data plays the most important role in retrieving information, either as textual annotations for multimedia data or as documents written in natural language. A crucial question is therefore: How to represent textual data in the Web to enable effective and efficient automatic processing to assist users. Information Retrieval (IR) researchers have developed various methods to help users find what they are looking for. To represent textual data, the Vector Space Model (VSM) [163] was developed. To ground the model in probabilistic theory, language models (LM) [147] evolved followed by advanced topic models (TM) [25] to represent latent topics within document collections. These models allow for dense representations of often sparse information and are applied in a multitude of applications, from machine learning to natural language processing, from recommender systems to information retrieval.

In this thesis we look at the different application domains and how language models and topic models can be applied beneficially.

1.1.1 Representation of Textual Data

Textual data needs to be represented in a suitable way to facilitate processing and analysing of information. These representations have to be efficient and effective with respect to computational resources and ideally the underlying models can be interpreted and understood easily by humans. We investigate two popular types of representing textual data: language models and topic models. Both have their advantages and disadvantages which we will shed light on in the context of different Web applications.

Language Models. Language models assign probabilities to words or word sequences based on probability distributions. Many different applications in natural language processing make use of language modeling, e.g. speech recognition, part-of-speech tagging, or machine translation [127]. The purpose of language models for these tasks is to capture the characteristics of a language and give a probability estimate for the next token in a sequence of observed input tokens.

In contrast to this type of use, language models are used in information retrieval to estimate the probability of a document to generate a particular query [126]. Therefore each document is represented by a language model, a multinomial distribution over words or n-grams. Often smoothing is applied to deal with unobserved test data and prevent zero probability estimates. Unigram models are most often used ignoring the order of words and leading to a simple bag-of-words representation. In this thesis we investigate the application of language models used in information retrieval to various related tasks and compare them with topic model representations.

Topic Models. The most common topic modelling approach is latent Dirichlet allocation (LDA). It is based on the hypothesis that a person writing a document has certain topics in mind. To write about a topic then means to pick a word with a certain probability from the pool of words of that topic. A whole document can then be represented as a mixture of different topics. When the author of a document is one person, these topics reflect the person's view of a document and her particular vocabulary. In the context of tagging systems, for example, where multiple users are annotating resources, the resulting topics reflect a collaborative shared view of the document and the tags of the topics reflect a common vocabulary to describe the document.

More generally, LDA helps to explain the similarity of data by grouping features of this data into unobserved sets. A mixture of these sets then constitutes the observable data. The method was first introduced by Blei et al. [25] and applied to solve various tasks including topic identification [72], entity resolution [15], and Web spam classification [18].

In this thesis we explore the use of language models and topic models for different Web applications, namely:

- *Filtering* of News Articles
- Text *Classification* Based on Sentiment
- Diversification of Web Search Results and Product Review *Rankings*
- Tag *Recommendation* in Folksonomies

In the following we will introduce the different application domains and elucidate the use of language models and topic models in these domains.

1.1.2 Language Models and Topic Models in Web Applications

We look at common tasks in the Web and how the different representations of textual data influence the solutions and results. We investigate four Web applications and how language models and topic models can be used in the different contexts.

Filtering. *Online news* is a major source of information for many people and has been steadily gaining popularity over printed news. Easy access worldwide and a nearly real-time availability of new breaking news are big advantages over classical paper-based news distribution. The downside of this success is the huge amount of news items generated everyday. For the common user, this situation presents new challenges, since the volume of news makes it difficult — if not impossible — to keep track of all important events.

The current solution is to look at news aggregator sites like Google News[1]. They offer an automatic clustering of news items into broader categories, and collect news from thousands of sources. This is done using information crawled from online news pages. They also offer a ranking based on the information in the Web. One challenge for such aggregators is to pick the most important stories as they break, and to feature them as soon as they are available. Traditional newspapers employ a team of editors to filter the most important news; automatic approaches need smart algorithms to predict such news. We present in Chapter 3 an approach to predict the importance of news items based solely on the content of an article. Besides using language models as textual representations, we also investigate the generation of an easy to understand representation for important news based on latent Dirichlet allocation. We will see how topic models can help to identify important topics using a set of training data and support vector machines to learn a classifier.

Classification. Sorting information according to (predefined) classes or categories allows users to quickly assess the characteristics of a dataset and/or select relevant information within a collection of heterogeneous documents. These classifications can be based on product categories in e-commerce portals, topical categories in online news providers, or spam/no-spam decisions in e-mail systems. Automatic text classification or categorization [127] plays an important role in the Web to structure information spaces.

In Chapter 4 we investigate the use of topic models to do a context-dependent classification of documents based on *sentiment orientation* [143]. More specifically, we propose an algorithm to assign sentiment scores to terms depending on their topical context. Assigning sentiment values to terms has been shown to be a non-trivial and time-consuming task for humans, and can be highly context-dependent (e.g. “scary movie” vs. “scary flight”). Sentiment lexica are useful for analyzing opinions in web collections (e.g., in blogs, news articles, product reviews, or comments), for context- and domain-dependent sentiment classification, and as sub-components of recommender systems.

We present different strategies for automatically generating topic-specific lexica from large corpora of review articles paired with user ratings. Our approach combines discriminative feature analysis techniques with latent topic extraction to infer the polarity of terms in a given topical context. Topic models allow in this setting to identify the topical context of a term and compute a sentiment score accordingly. We evaluate our approach based on rating prediction as well as on direct user assessment showing improvement over state-of-the-art systems. We further show the viability of our techniques towards complementing existing sentiment lexicon generation approaches.

[1] `http://news.google.com/`

Ranking. One of the main tasks in Information Retrieval [9, 126] is to find relevant information based on an information need of a user often expressed with a keyword query. The results are rankings ordered according to a particular objective. The most prominent rankings are Web search results. Here various factors need to be weighted in order to rank the Web pages. For example, popularity, recency, relevance, etc. need to be considered. Rankings of product reviews need to take other factors into account like product ratings or sentiments expressed in the review. In both cases, diversity of the results is not to be neglected.

Search engine results are often biased towards a certain aspect of a query or towards a certain meaning for ambiguous query terms. Diversification of search results offers a way to supply the user with a better balanced result set increasing the probability that a user finds at least one document suiting her information need. In Section 5.2, we present a reranking approach based on minimizing variance of *Web search results* to improve topic coverage in the top-k results. We investigate two different document representations as the basis for reranking. Smoothed language models and topic models derived by latent Dirichlet allocation. To evaluate our approach we used the TREC sub-topic evaluation and a manually labeled dataset on the one hand, and on the other hand, we selected 240 queries from Wikipedia disambiguation pages. This provides us with ambiguous queries together with a community generated balanced representation of their (sub-)topics. For these queries we crawled two major commercial search engines. Our results show that minimizing variance in search results by reranking relevant pages significantly improves topic coverage in the top-k results with respect to Wikipedia, and gives a good overview of the overall search result. Moreover, latent topic models achieve competitive diversification with significantly less reranking.

E-commerce Web sites owe much of their popularity to *consumer reviews* provided together with product descriptions. On-line customers spend hours and hours going through heaps of textual reviews to build confidence in products they are planning to buy. At the same time, popular products have thousands of user-generated reviews. Current approaches to present them to the user or recommend an individual review for a product are based on the helpfulness or usefulness of each review. In Section 5.3 we look at the top-k reviews in a ranking to give a good summary to the user with each review complementing the others. To this end we use latent Dirichlet allocation to detect latent topics within reviews and make use of the assigned star rating for the product as an indicator of the polarity expressed towards the product and the latent topics within the review. We present a framework to cover different ranking strategies based on the user's need: Summarizing all reviews; focus on a particular latent topic; or focus on positive, negative or neutral aspects. We evaluated the system using manually annotated review data from a commercial review Web site.

Recommendation. More and more content on the Web is generated by users. To organize this information and make it accessible via current search technology, tagging systems have gained tremendous popularity. The most popular tagging sites like Flickr[2], LastFm[3], YouTube[4], or Delicious[5] provide support for users to find information and organize their own content. Therefore these sites allow users to annotate content with their own keywords (tags), opening up the possibility to retrieve content using traditional keyword

[2] Flickr: `http://www.flickr.com`

[3] LastFm: `http://www.last.fm`

[4] YouTube: `http://www.youtube.com`

[5] Delicious: `http://delicious.com`

search. Especially multimedia content like music, photos, or videos rely on manually added meta information. Adding keywords to content (tagging) is the only feasible way to organize multimedia data at that scale and to make it searchable. These keywords can be freely chosen by a user and are not restricted to any taxonomy. This results in some benefits like flexibility, quick adaption, and easy usability, but has also some drawbacks.

To support the user in choosing the right keywords, tag recommendation algorithms have emerged. In this setting, not only the content is decisive for recommending relevant tags but also the user's preferences. In Chapter 6 we introduce an approach to tag recommendation in *folksonomies* based on language models and topic models. We first investigate the use of latent Dirichlet allocation for unpersonalized, collective tag recommendation and then show how the combination of probabilistic models of tags from the resource with tags from the user can improve personalized tag recommendation. Extensive experiments on real world datasets crawled from a big tagging system show that collective tag recommendation using LDA improves recall for searching in tagging systems and personalization improves tag recommendation in the classic leave-on-out evaluation setting. Our approaches also significantly outperform state-of-the-art approaches for tag recommendation.

1.2 Outline of the Thesis

In Chapter 2 we look at the foundations and background of language models and topic models. The four following chapters present the work in the different application domains, namely Chapter 3 describes our findings for filtering news articles based on importance. Results from this chapter were published in:

- Ralf Krestel and Bhaskar Mehta. Learning the Importance of Latent Topics to Discover Highly Influential News Items. In *KI 2010: Advances in Artificial Intelligence, 33rd Annual German Conference on AI, Karlsruhe, Germany, September 21-24, 2010. Proceedings*, volume 6359 of *Lecture Notes in Computer Science*, pages 211–218, Berlin, Heidelberg, September 21–24 2010. Springer. BEST PAPER AWARD

- Ralf Krestel and Bhaskar Mehta. Predicting News Story Importance using Language Features. In *WI-IAT '08: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 683–689, Washington, DC, USA, December 9–12 2008. IEEE Computer Society

In Chapter 4, we explore the possibility to generate a sentiment lexicon using topic models for classifying documents based on sentiment. Chapter 5 discusses two applications to diversify result rankings: The first in the context of product review rankings, and the second in the context of Web search result ranking. The results were published in:

- Ralf Krestel and Peter Fankhauser. Web Search Result Diversification by Reranking. *Information Retrieval*, 2012. Springer. Amsterdam, Netherlands. (in press)

- Ralf Krestel and Nima Dokoohaki. Diversifying Product Review Rankings: Getting the Full Picture. In *WI-IAT '11: Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Washington, DC, USA, August 22–27 2011. IEEE Computer Society. BEST PAPER AWARD

In Chapter 6 we look at collective tag recommendation with results published in:

- Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent Dirichlet Allocation for Tag Recommendation. In *Proceedings of the 2009 ACM Conference on Recommender Systems (RecSys 2009)*, pages 61–68, New York City, New York, USA, October 23–25 2009. ACM

- Ralf Krestel and Peter Fankhauser. Tag recommendation using probabilistic topic models. In *ECML/PKDD Discovery Challenge (DC'09), Workshop at ECML/PKDD 2009*, pages 131–141, September 7th 2009

- Ralf Krestel and Ling Chen. The Art of Tagging: Measuring the Quality of Tags. In *ASWC '08: Proceedings of the 3rd Asian Semantic Web Conference on The Semantic Web*, volume 5367 of *Lecture Notes in Computer Science*, pages 257–271, Berlin, Heidelberg, February 2–5 2009. Springer-Verlag

And we look at personalized tag recommendation with results published in:

- Ralf Krestel and Peter Fankhauser. Personalized topic-based tag recommendation. *Neurocomputing*, 76(1):61–70, 2012. Elsevier. Amsterdam, Netherlands

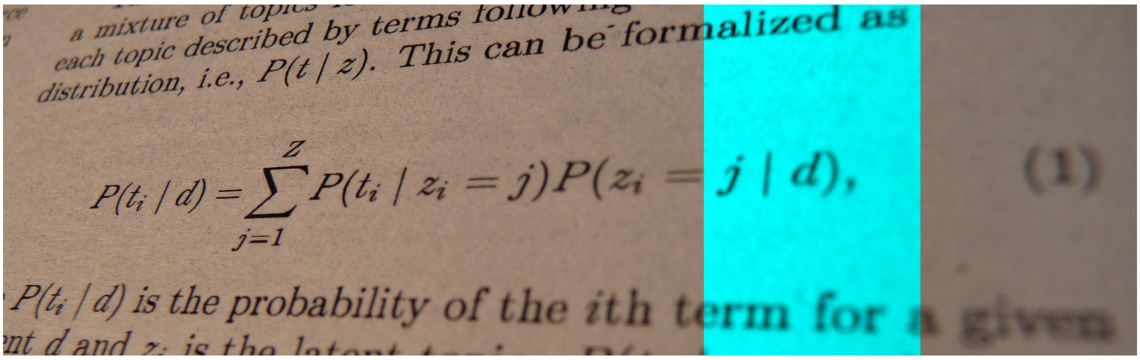
- Ralf Krestel and Peter Fankhauser. Language Models & Topic Models for Personalizing Tag Recommendation. In *WI-IAT '10: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 82–89, Washington, DC, USA, August 31–September 3 2010. IEEE Computer Society

We conclude with a summary and outlook in Chapter 7 describing possible future applications for topic modelling and possible future work for the presented application domains.

While working on this thesis, I contributed to various other publications not directly linked to this work but illustrating the context from which this thesis originated. These are a book chapter [197], two journal articles [200, 51], a couple of conference papers [17, 198, 49, 50, 101, 114, 116], as well as some workshop papers [117, 102, 196, 103, 100, 199] and some posters [105, 16, 115].

CHAPTER
TWO

FOUNDATIONS AND BACKGROUND



a mixture of topi
each topic described by terms followin
distribution, i.e., $P(t \mid z)$. This can be formalized as

$$P(t_i \mid d) = \sum_{j=1}^{Z} P(t_i \mid z_i = j) P(z_i = j \mid d), \qquad (1)$$

$P(t_i \mid d)$ is the probability of the ith term for a given
nt d and z_i is the latent

2.1 Introduction

Understanding natural language is one of the key tasks in artificial intelligence research. Grasping the meaning of texts is in general still impossible for computers. Several advancements over the last decade in related areas such as Semantic Web, Machine Learning, Information Retrieval, or Natural Language Processing (NLP) have helped to get a step closer to automatic understanding the meaning of texts. For specific domains and tasks, the development of ontologies, wikis, sentiment classification, semantic parsers, Web 2.0, etc. has led to systems that are able to process textual data similar to humans (cf. IBM's Watson [59] for question-answering).

An important part in automatic processing of natural language is the representation of textual data. In the context of Information Retrieval (IR), simple language models[1] (LM) have been used successfully to perform basic keyword search. In the Machine Learning (ML) context, bag-of-words approaches are often superior to more complex representations when it comes to document classification, for example. We will discuss language models as a mean to represent textual data in Section 2.2.

If more semantic knowledge is needed, latent variables are introduced into the model to cover more complex relations between words. One early approach to this end is latent semantic analysis (LSA) [47] followed by probabilistic latent semantic analysis (PLSA) [83] which introduced explicit latent topics. Latent Dirichlet allocation (LDA) [25] is a further development of PLSA modeling documents using latent topics. We will have a closer look at these topic models in Section 2.3.

[1] Usually based on unigrams (Bag of words model)

2.2 Language Models

Language models (LM) were introduced in the context of speech recognition to complement the acoustic models and compute probabilities for word sequences [94]. The idea of modeling language probabilistically is even older, dating back to Markov and Shannon [82].

Besides speech recognition, machine translation, part-of-speech tagging, and other NLP tasks, information retrieval makes heavily use of language models [4]. Finding relevant documents given a query has been formalized as a language modeling problem to improve search [147, 170, 131].

In this thesis, we make use of simple language models within different Web applications. We are mainly interested in unigram language models but also make use of n-gram models with $n > 1$. Since unigrams might not give an accurate picture of what a document is about, we can also extract n-grams of variable lengths (multigrams). This is interesting in domains where multi-term phrases might be important to model the underlying data, e.g. "Microsoft Windows 7 Professional", or "not recommended" in product review data. Based on the work of Deligne and Bimbot [48], we can compute multigram models for documents in a corpus the following way: Each sentence is considered as a sequence of n-grams with variable length. The likelihood of a sentence is computed by summing up the individual likelihoods of the n-grams corresponding to each possible segmentation of the sentence. This is done using a Viterbi-like algorithm to find the maximum likelihood segmentation. In an iterative fashion, we re-estimate and update the probabilities until convergence. For unigram language models counting occurrences of single terms is enough to compute the language model.

In its simplest form, a language model for a document d with words w_i can be formalized using a maximum likelihood estimate:

$$P_{ml}(w \mid d) = \frac{c(w,d)}{\sum_{w_i \in d} c(w_i,d)} \tag{2.1}$$

where $c(w,d)$ is the count of word w in document d.

This very basic language model is often used in machine learning to represent textual data and train a classifier. Often additional weights are introduced for each word, such as tf*idf scores. Especially for vector space models (VSM) this approach is called a *bag-of-words* (BOW) representation. In the context of user modeling, simple language models are used to represent *profiles* of users. We will refer to language models independent of the application area.

Smoothing To prevent the probability $P_{lm}(w \mid d)$ from being zero in case word w does not appear in document d, various smoothing methods have been introduced and compared [208]. The unsmoothed model using a maximum likelihood estimate shown in Equation 2.1 can be complemented by different components, with the Laplace smoothing being the simplest, adding 1 to each count:

$$P_{lp}(w \mid d) = \frac{c(w,d)+1}{\sum_{w_i \in d} c(w_i,d)+1} \tag{2.2}$$

The Jelinek-Mercer smoothing adds a corpus component and combines it linearly with P_{ml}:

$$P_{jm}(w \mid d) = (1-\lambda)P_{ml}(w \mid d) + \lambda P_{ml}(w \mid C) \tag{2.3}$$

where C is the collection or corpus.

Dirichlet smoothing extends the Jelinek-Mercer smoothing by taking document lengths into account:

$$P_{di}(w \mid d) = \frac{c(w,d) + \mu P(w \mid C)}{\sum_{w_i \in C} c(w_i,d) + \mu} \tag{2.4}$$

We will make use of smoothing methods when necessary in the different applications.

2.3 Topic Models

Topic models are generative models that assume that different topics can be found in a collection of documents. They are based on the hypothesis that a person writing a document has certain topics in mind. To write about a topic then means to pick a word with a certain probability from the pool of words of that topic. A whole document can then be represented as a mixture of different topics. When the author of a document is one person, these topics reflect the person's view of a document and her particular vocabulary.

This explanation of the generative process of creating a document form the basis of reverse engineering the latent topics from a collection. Statistical methods are used to find the model with the highest probability for generating the document collection. In the following we will look in more detail into Probabilistic latent semantic analysis (PLSA) and latent Dirichlet allocation (LDA).

2.3.1 Probabilistic Latent Semantic Analysis (PLSA)

Generative topic models have been developed after methods like latent semantic analysis (LSA) [47] allowed to reduce the dimensionality of textual data. The vector space model (VSM) used before, not only has a dimensionality problem, but also doesn't capture synonyms, hypernyms, or polysemy.

In Information Retrieval, weighting schemes like tf*idf can be used to extract salient terms in documents and reduce the amount of terms to consider. But the dimensionality problem still exists. LSA reduces the dimensionality by projecting the entries in the document-term matrix into a subspace. Singular value decomposition is used to represent each document as a linear combination within this subspace.

A probabilistic, generative grounding of the LSA method [145] was needed to explain the results of LSA. To this end PLSA [83] was developed. A mixture decomposition method derived from a latent class model was proposed having a solid grounding in statistics in comparison to LSA which has its grounding in linear algebra. PLSA uses a maximum likelihood model and expectation maximization.

Later, LDA was developed to overcome some of the shortcomings of PLSA, namely not providing a probabilistic model on document level. Girolami and Kabán [65] then showed that PLSA can be seen as a special case of LDA.

2.3.2 Latent Dirichlet Allocation (LDA)

LDA helps to explain the similarity of data by grouping features of this data into unobserved sets. A mixture of these sets then constitutes the observable data. The method was first introduced by Blei et al. [25] and applied to solve various tasks including topic identification [72], entity resolution [15], and Web spam classification [18].

Figure 2.1 shows the plate notation for latent Dirichlet allocation. LDA identifies a given number of $|Z|$ topics within a corpus of $|D|$ documents. Each term t in a review with N_d terms is associated with a topic z. Being the most important parameter for LDA,

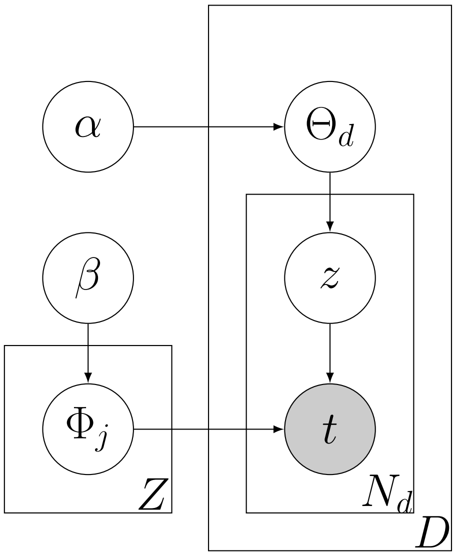


Figure 2.1: Plate notation for latent Dirichlet allocation

the number of latent topics $|Z|$ determines the granularity of the resulting topics, as we will see later. This number has to be fixed in advance and depends on the application of the topic model and the underlying corpus [6, 71]. In order to find the latent topics, LDA relies on stochastic modeling.

The modeling process of LDA can be described as finding a mixture of topics for each resource, i.e., $P(z \mid d)$, with each topic described by terms following another probability distribution, i.e., $P(t \mid z)$. This can be formalized as

$$P(t_i \mid d) = \sum_{j=1}^{Z} P(t_i \mid z_i = j) P(z_i = j \mid d), \tag{2.5}$$

where $P(t_i \mid d)$ is the probability of the ith term for a given document d and z_i is the latent topic. $P(t_i \mid z_i = j)$ is the probability of t_i within topic j. $P(z_i = j \mid d)$ is the probability of picking a term from topic j in the document. The number of latent topics Z has to be defined in advance and allows to adjust the degree of specialization of the latent topics. LDA estimates the topic–term distribution $P(t \mid z)$ and the document–topic distribution $P(z \mid d)$ from an unlabeled corpus of documents using Dirichlet priors for the distributions and a fixed number of topics.

Inference and Parameter Estimation Gibbs sampling [72, 172] is one possible approach to estimate the parameters: It iterates multiple times over each term t_i in document d_i, and samples a new topic j for the term based on the probability $P(z_i = j|t_i, d_i, z_{-i})$ based on Equation 2.6, until the LDA model parameters converge.

$$P(z_i = j \mid t_i, d_i, z_{-i}) \propto \frac{C^{TZ}_{t_i j} + \beta}{\sum_t C^{TZ}_{tj} + T\beta} \frac{C^{DZ}_{d_i j} + \alpha}{\sum_z C^{DZ}_{d_i z} + Z\alpha} \tag{2.6}$$

C^{TZ} maintains a count of all topic–term assignments, C^{DZ} counts the document–topic assignments, both excluding the current assignment of z_i for term t_i, z_{-i} represents all topic–term and document–topic assignments except the current assignment z_i for term t_i. And α and β are the (symmetric) hyperparameters for the Dirichlet priors, serving as smoothing parameters for the counts. Based on the counts the posterior probabilities in Equation 2.5 can be estimated as follows:

$$P(t_i \mid z_i = j) = \frac{C^{TZ}_{t_i j} + \beta}{\sum_t C^{TZ}_{tj} + T\beta} \tag{2.7}$$

$$P(z_i = j \mid d_i) = \frac{C^{DZ}_{d_i j} + \alpha}{\sum_z C^{DZ}_{d_i z} + Z\alpha} \tag{2.8}$$

Other methods for inference and parameter estimations are variational Bayes approximations [25], collapsed variational Bayes inference [175], or expectation propagation [132]. The connections and empirical comparisons of different approaches is described in [7].

Implementations There exist various implementations of LDA in Java, C, Python, or Mathlab:

- Blei `http://www.cs.princeton.edu/~blei/lda-c/`
- Block `http://www.bradblock.com/tm-0.1.tar.gz`
- Buntine `http://www.nicta.com.au/people/buntinew/discrete_component_analysis`
- Heinrich `http://arbylon.net/projects/knowceans-ilda/knowceans-ilda.zip`
- LingPipe `http://alias-i.com/lingpipe`
- McCallum `http://mallet.cs.umass.edu/`
- Nallapati `https://sites.google.com/site/rameshnallapati/software`
- Phan and Nguyen `http://gibbslda.sourceforge.net/`
- Ramage and Rosen `http://nlp.stanford.edu/software/tmt/tmt-0.3/`
- Řehůřek `http://nlp.fi.muni.cz/projekty/gensim/`
- Steyvers and Griffiths `http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm`

In this thesis we used the LDA implementations of LingPipe and McCallum's Mallet package. They both use Gibbs sampling to discover topics.

2.3.3 Extensions of Latent Dirichlet Allocation

Various extensions to the basic LDA model have been proposed. To model topic changes over time and the evolution of topics in time-stamped corpora, dynamic topic models [23] were introduced. State space models on the LDA parameters model the evolution of the topics. An online approach using multiple timescales to model the topic evolution is presented in [90].

In the context of prediction and classification, supervised machine learning techniques are incorporated to predict document labels for unseen documents. These supervised topic models [24, 158] allow to model not only the textual data of a document but also an observable label. Besides topics, authors can also be modeled as multinominal distributions. In this case the basis are the topics not the terms [161].

Combining topics and aspects in a joint model has been proposed in [180]. Especially for product review data this allows to model products and features of products individually. Topics and aspects are also modeled in [146] to discover aspects or viewpoints in general documents.

Concerning the relation between topics, hierarchical topic models [21, 174] and correlated topic models [22] were developed. To get a hierarchy of topics, the nested Chinese restaurant process is used. To find topics that are related with each other, a logistic normal distribution is used within the LDA model. A generalization combining hierarchies and correlation was developed called Pachinko Allocation Model (PAM) [119] using a directed acyclic graph (DAG) to model arbitrary relations between topics and terms.

2.3.4 Evaluation of Topic Models

The problem of evaluating and comparing different topic models with respect to quality and efficiency is not trivial to solve. One approach is to evaluate the topic model by its performance on a secondary task such as document classification or information retrieval.

An approach to evaluate topic coherence using WordNet, Wikipedia, or Google is presented in [136] indicating that pointwise mutual information (PMI) on Wikipedia documents, as well as coocurrence of terms in titles of documents returned by Google achieve similar results as human annotators.

The most commmon way for topic model evaluation is to estimating the probability of held-out test documents for a topic model build on training documents. This offers a direct way to compare the quality of different topic models [186]. Since this probability computation is intractable, different estimation algorithms are used. Since the results often do not correlate with human annotators [32], manual human judgement offers an alternative method. A comparison of LDA generated topics with human topic construction can be found in [31].

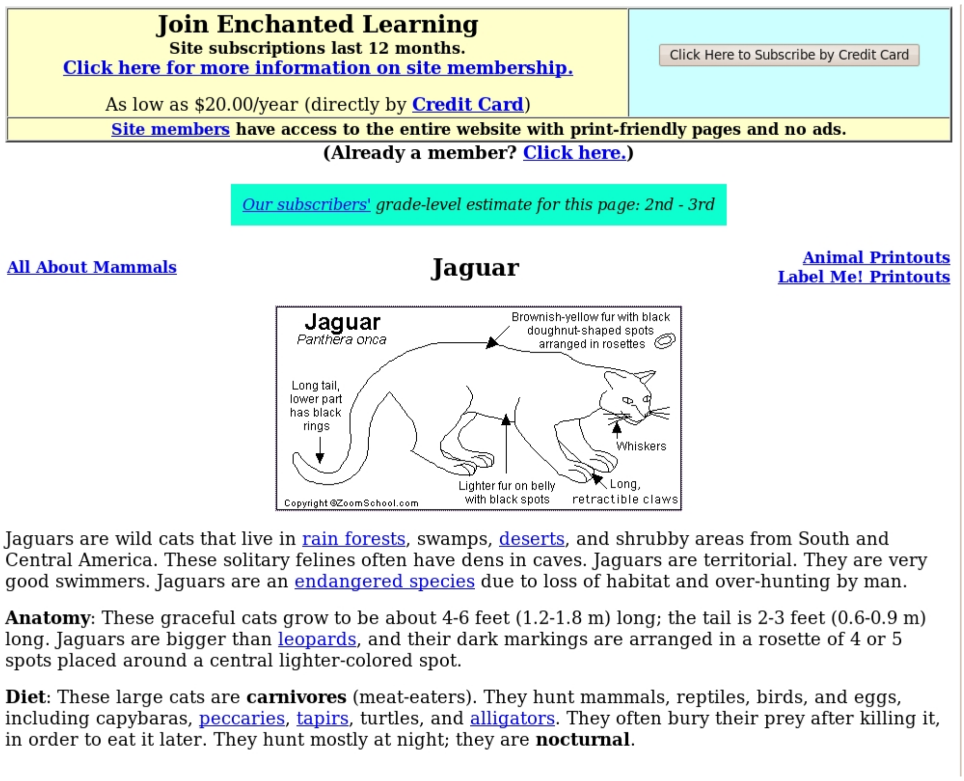
2.4 Use Case: Language Model and Topic Model Example

To illustrate the process of generating a language model and a topic model for a particular document we look at an example page from the Web. It appeared in the top 100 results in Google when queried with the term "jaguar". The most prominent interpretations of the term "jaguar" comprise the cat and the car manufacturer. We picked a Web page about the cat and look at the computed language models and topic models.

1. Original Document. The starting point of the modeling process is the original HTML page as rendered by a Web browser. It may contain images, text, graphics, or even sound and videos. For our example Web page, a screenshot is shown in Figure 2.2 displaying visual information as well as some textual information.

2. Extracted Textual Data. To get a representation of the textual content of the pages we need to extract the fulltext and remove templates or boilerplate information. Also images, videos, or advertisements are removed. The extracted textual data for the "jaguar" (cat) example page is shown in Figure 2.3.

3. Bag-of-Words Representation. After removing common stop words we can optionally apply stemming or lemmatization. In our examples, we only removed common stop words and ordered the terms alphabetically. Since we are interested in unigram language models the context of each word or the correct ordering is neglected. Also our topic model does not care about the ordering. The results of this step is shown in Figure 2.4. As can be seen, the boilerplate detection was not 100 percent accurate, identifying terms like



Join Enchanted Learning

Site subscriptions last 12 months.

Click here for more information on site membership.

As low as $20.00/year (directly by **Credit Card**)

Site members have access to the entire website with print-friendly pages and no ads.

(Already a member? Click here.)

Our subscribers' grade-level estimate for this page: 2nd - 3rd

All About Mammals

Jaguar

Animal Printouts
Label Me! Printouts

Jaguar
Panthera onca
Brownish-yellow fur with black doughnut-shaped spots arranged in rosettes
Long tail, lower part has black rings
Whiskers
Lighter fur on belly with black spots
Long, retractible claws
Copyright ©ZoomSchool.com

Jaguars are wild cats that live in rain forests, swamps, deserts, and shrubby areas from South and Central America. These solitary felines often have dens in caves. Jaguars are territorial. They are very good swimmers. Jaguars are an endangered species due to loss of habitat and over-hunting by man.

Anatomy: These graceful cats grow to be about 4-6 feet (1.2-1.8 m) long; the tail is 2-3 feet (0.6-0.9 m) long. Jaguars are bigger than leopards, and their dark markings are arranged in a rosette of 4 or 5 spots placed around a central lighter-colored spot.

Diet: These large cats are **carnivores** (meat-eaters). They hunt mammals, reptiles, birds, and eggs, including capybaras, peccaries, tapirs, turtles, and alligators. They often bury their prey after killing it, in order to eat it later. They hunt mostly at night; they are **nocturnal**.

Figure 2.2: Main part of the Web page `http://www.enchantedlearning.com/subjects/mammals/cats/jaguar/Jaguarprintout.shtml`

EnchantedLearning.com is a user-supported site. As a bonus, site members have access to a banner-ad-free version of the site, with print-friendly pages. Our subscribers' grade-level estimate for this page: 2nd - 3rd

Jaguars are wild cats that live in rain forests, swamps, deserts, and shrubby areas from South and Central America. These solitary felines often have dens in caves. Jaguars are territorial. They are very good swimmers. Jaguars are an endangered species due to loss of habitat and over-hunting by man.

Anatomy: These graceful cats grow to be about 4-6 feet (1.2-1.8 m) long; the tail is 2-3 feet (0.6-0.9 m) long. Jaguars are bigger than leopards, and their dark markings are arranged in a rosette of 4 or 5 spots placed around a central lighter-colored spot.

Diet: These large cats are carnivores (meat-eaters). They hunt mammals, reptiles, birds, and eggs, including capybaras, peccaries, tapirs, turtles, and alligators. They often bury their prey after killing it, in order to eat it later. They hunt mostly at night; they are nocturnal.

Enchanted Learning Search

Advertisement. Advertisement. Advertisement.

Figure 2.3: Fulltext for Jaguar (cats) Web page

2nd 3rd access advertisement advertisement advertisement alligators america anatomy areas arranged banner bigger birds bonus bury capybaras carnivores cats cats cats caves central central colored dark dens deserts diet due eat eaters eggs enchanted enchantedlearning endangered estimate feet feet felines forests friendly good graceful grade grow habitat hunt hunt hunting including jaguars jaguars jaguars jaguars killing large learning leopards level lighter live long long loss mammals man markings meat members night nocturnal order pages peccaries placed prey print rain reptiles rosette search shrubby solitary south species spot spots subscribers supported swamps swimmers tail tapirs territorial turtles user version wild

Figure 2.4: Bag-of-Words for Jaguar (cats) Web page

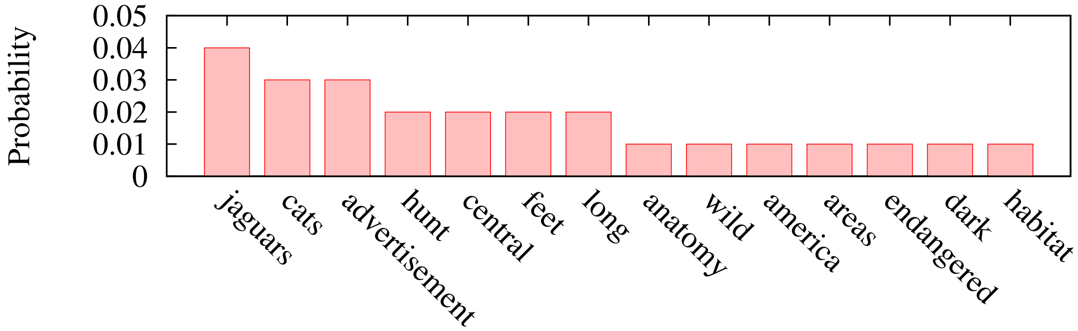


Figure 2.5: Language model for Jaguar (cats) Web page (top terms)

“advertisement” as part of the fulltext content. This Bag-of-Words representation of the documents serve as the input for the language model generation as well as for the latent Dirichlet allocation process to generate the topic model.

4a. Language Model The language model of our jaguar Web page can be seen in Figure 2.5. In our example we use a corpus consisting of the top 500 search results retrieved by the Google search engine. This corpus information is needed for different smoothing methods that could be applied to the language model. For example, a term not appearing in the jaguar page but in all other pages in the corpus might get a higher probability in the jaguar language model than a term appearing in only one other document.

4b. Topic Model. To compute the latent topics we take the bag-of-words representation of all documents in our corpus. In our example we chose to extract 50 topics. The multinominal distribution of these topics for our jaguar web page can be seen in Figure 2.6. Only the top 10 most likely topics are displayed.

In Figures 2.7, 2.8, and 2.9 the multinominal distributions of terms for topics 28, 2, and 3 are shown. They cover different meanings and aspects of the term jaguar. Topic 28 clearly relates to cats, topic 2 describes aircrafts, and topic 3 car dealerships.

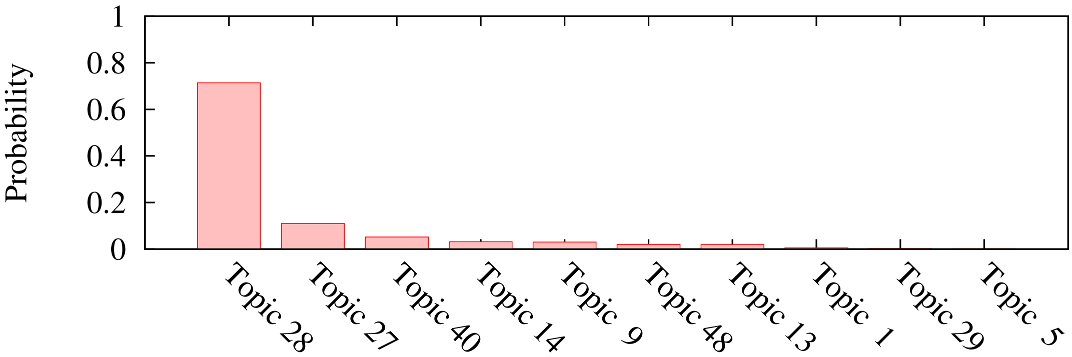


Figure 2.6: Topic model for jaguar (cats) Web page (top topics)

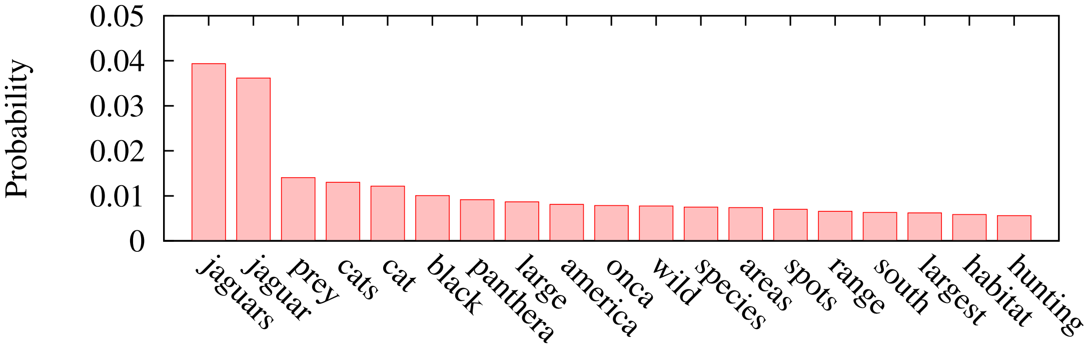


Figure 2.7: Topic 28 for jaguar Web pages (top terms)

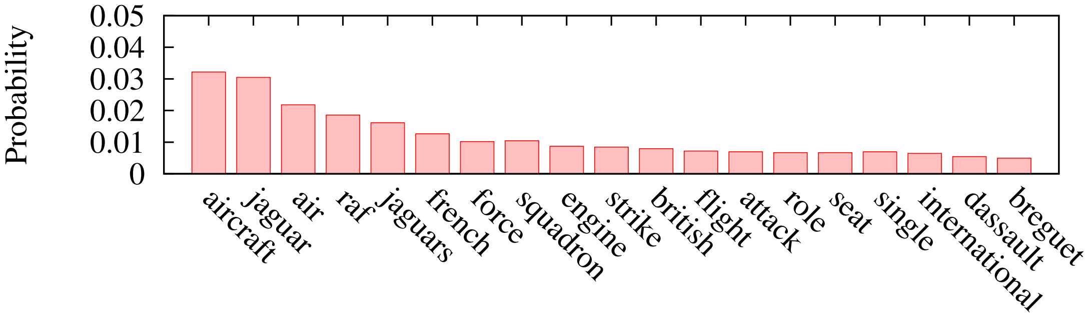


Figure 2.8: Topic 2 for jaguar Web pages (top terms)

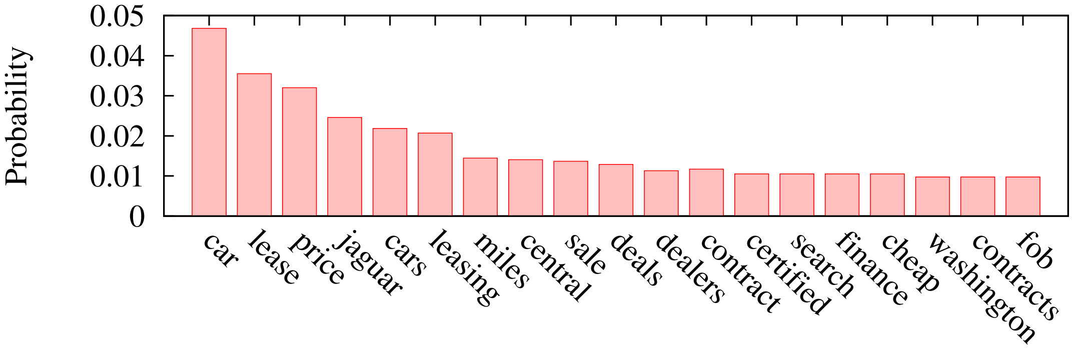


Figure 2.9: Topic 3 for Jaguar Web pages (top terms)

CHAPTER

THREE

NEWS FILTERING

In this chapter we investigate the use of language models and topic models to identify important news articles. We train a support vector machine using either textual features or latent features as input and analyze the resulting classifiers.

3.1 Introduction

Online news is a fast growing source of information for people around the world. According to surveys by Nielson/NetRatings, online news readership grew at 47% in 2004 and more than 40% in 2005. Importantly, many users have indicated online news as their primary source (47%) compared to conventional news sources like radio (16%), TV (18%) and printed newspapers (12%). *Pew Internet*[1] also reports a robust growth in blog readership, doubling in one year to 32 million at the end of 2005. 5% of responders also reported using RSS feeds, thus combining multiple sources of information; this number is also forecasted at 50% growth for the next few years. All these trends point to impending information overflow, where automatic support for filtering information is important.

Given that users increasingly prefer online sources, mainly due to their 24-hour presence and quick turn around time, people are also keen to learn about important events as soon as they happen. This is especially important for events with economic or political implications. Economics researchers have found a high correlation of news items and changes in stock market prices as a result [70]. Various stock market trading houses have in-house news analysts who model the impact of a news item. For such companies and their employees, filtering important news and predicting if they will become important enough for the market to take notice, is of paramount importance. Electronic news information systems

[1] www.PewInternet.com

which notice such news immediately are thus highly desirable; input from these systems can further be tied into financial prediction models.

Newspapers or online news providers present news in an unpersonalized way. They have editors who pick the most important news stories for their readers. Good newspapers do this in an unbiased and diverse way. The newspaper reader will constantly be confronted with new topics, new opinions, and new views, allowing him to broaden his knowledge and to stay informed on important events. The advantages of an edited newspaper are lost if any kind of personalization is employed. Personalization leads to a limited world view that only covers a focused set of topics or events and in the worst case only a certain opinion about a particular topic. A personalized newspaper does not serve the purpose of a general newspaper anymore. There is no more "surprise" for the user. Important topics are filtered out if they do not fit into the previous reading patterns of a user. Instead of getting controverse view points, articles containting the same topics or opinion as the reader has seen before are prefered. Imagine a user in favour of the democratic party who reads a personalized newspaper. She wants to inform herself for the upcoming elections. In the worst case, the personalization system knows about the pro-democratic attitude of the user and only presents pro democratic articles. Diversity or controversal news coverage is not supported.

Nevertheless, filtering of news articles is essential, solely because of the huge amount published every day. But based on the previous paragraphs we argue that it should not be based on personalization but on importance in an unbiased and highly objective way. Even though it is difficult to draw a sharp line between important news and unimportant ones, it seems easy to identify extremely important news and really unimportant news just by looking at the number of news providers covering a given story. Another alternative is to look at user feedback, e.g. click through rates. While these social features are very strong indications, they are often known only after the story has been around for a while. Our aim is to examine news stories without such signals, so that a reasonably accurate prediction can be made as soon as the article is available electronically. The source of a news story can still be used as signal since this information is available at publishing time. However, the news industry today sources news stories from aggregated news agencies (e.g. the associated Press, or Reuters), and clearly not every story can be deemed important. The filtering algorithm we want to devise relies on plain text, and should make the job of human editors easier by picking the most important stories first. This approach can also be imagined in a TV scenario, where transcribed text from the TV audio is run through a classifier and can recognize important stories as they are made available.

From an historical point of view, certain types of events have triggered the creation of many news articles all over the world. Events like the breakout of a war or big natural catastrophes are important in the sense that they get global news coverage. Our goal is to predict the number of newspapers who will pick up a certain topic and thus estimate the importance of this topic.

We consider in this chapter a learning system which identifies crucial factors that make a news item important. Note that such factors may not be directly observed, or be directly mapped to observed signals. In particular, we consider a regression setup to predict the importance of a new news item. Further our approach is content-based; our goal is to find decisive natural language features which make a newspaper article more important than another. We use several natural language signals extracted from news items, and do not rely on social feedback (e.g. Digg); this is important since user feedback, though very helpful for prediction, is associated with high latency, and sufficient user feedback may not be available when information is still new. In this chapter, we try to find objective measures for the importance of an article without regarding personal preferences of the user or other biased subjective views.

We first look at language models to represent news and predict importance and then, the generation of an easy to understand representation for important news based on LDA topics is presented. A comparison between the two approaches shows improvements in accuracy using topic models over plain language models.

Problem Statement and Description. We consider a news item as a collection of text, with additional attributes for title and source. Consider that news is written in language with vocabulary $\mathcal{V}$ of size $M = |\mathcal{V}|$; the i^{th} word in the vocabulary is represented by v_i. We use the standard bag-of-words representation to build a language model, which compresses a document $\mathcal{D}$ into a vector d, such that

$$\mathsf{d} = \{f_1, f_2, \cdots, f_M\} \ , \tag{3.1}$$

where f_i is the frequency of the word v_i in document d.

Given a collection $\mathbf{C}$ of N document-vectors (features) $\{\mathsf{d}_1, ..., \mathsf{d}_N\}$, we procure *labels* representing importance from a qualified source. Such labels are quantitative, either continuous (say, between 0 to 1), or discrete (say, 1–4). Thus, we have a supervised regression setup: $\{(\mathsf{d}_1, l_1), ..., (\mathsf{d}_N, l_N)\}$. We are interested in learning an Importance Classifier function $\mathcal{I}$ such that

$$\mathcal{I}(\mathbf{C}) \rightarrow \{l\}^{|\mathbf{C}|} \tag{3.2}$$

where l is the set of labels. Specifically, we would like to learn a classifier with the lowest classification error over a training set with known labels $\mathbf{L}_{\mathcal{C}}$:

$$\mathcal{I}^* = \underset{\mathcal{I}}{\operatorname{argmin}} \sum_{i \in \mathbf{L}_{\mathcal{C}}} |\mathcal{I}(i) - \mathbf{L}_{\mathcal{C}}(i))|^2 \tag{3.3}$$

We also explore other classifiers which use features sets other than $\mathcal{C}$; examples include only nouns, verbs, adjectives etc. Note further that using the kernel trick, we can project the news items into feature space. We use several sets of extracted features and explore the predictive performance of these features.

In this setting, each feature-vector represents one news topic. Features could be the words of the articles to be represented. This language model based approach, has however certain drawbacks. Firstly, the huge size of the feature space; secondly, the sparsity of the data; and thirdly the interpretability of the model built by the classifier. To overcome these issues, we make use of an intermediate step to represent news topics using latent Dirichlet allocation for feature reduction and easily interpretable latent topics extracted from a training corpus. To evaluate the effectiveness of LDA topics as features for importance prediction we compare our approach with alternative feature extraction methods.

3.2 Language Models and Topic Models for News Importance Prediction

In this section we present our approach to predict the importance or impact news articles might have. We train a support vector machine (SVM) based on a bag-of-words approach and an approach based on latent topics. We analyze the features to identify the most salient ones. To train and test the classifier we use news and statistics from a news aggregator site. We show that the language model representation of news articles can be easily implemented and the single terms can be directly used as input features for machine learning algorithms. We further analyze the latent topics used as input features for the support vector machine and show how LDA can be used to improve accuracy and provide insight into importance of topics.

3.2.1 Approach

Our proposed model for news ranking is exploratory in nature, that means, we are interested in finding which features are most indicative of importance. Traditionally, tree classifiers have been used for such a supervised classification setup. However, support vector machines (SVM) [184] are the de facto method used for classification nowadays. This is because SVMs can be much more accurate, and use non-linear regression as well by using kernels. A drawback of SVMs is the lack of a directly interpretable model; one can however interpret the importance of a feature, by looking at its weight in the support vector learnt. Thus, good feature selection is the key to our solution. We will look into SVMs briefly and then, as established before, we describe features of news which are textual in nature and we could use as input for our SVM followed by a description of the latent features extracted using latent Dirichlet allocation.

Support Vector Machine. To build a support vector machine, we need training data. This data needs to be labeled with the class variable. This can be a multi nominal value indicating a multi-class classification problem or a continuous value indicating a regression problem. We look at both cases using for example two classes "important" and "unimportant" to describe a news article, or a numeric value to indicate the importance of a news item. Each training instance is represented as a data point in a vector space model. The dimensions of this vector space are the input features. If the input features are the words occurring in a news article, each distinct word represents one dimension. The goal of the SVM is now to find a hyperplane in this vector space that separates the training points of two different classes. The distance of the training points closest to the hyperplane is maximized in order to provide good classification performance for unknown test instances and reduce the generalization error. To construct the hyperplane, not all training points need to be considered. Only the ones close to the hyperplane influence their position. The vectors for these instances are called support vectors and are needed to describe the hyperplane mathematically. If the two classes are not linearly separable in the vector space, then there exists no hyperplane that divides the two classes accurately. In these cases, the kernel-trick is used to transform the data into a higher dimensional space where the two classes are linearly separable.

Part-of-Speech Tagging. Since the news space consists of free-form text, we can have scalability problems in dealing with large vocabularies. We therefore introduce part-of-speech (POS) tagging: a POS tagger labels words in a sentence as a noun, verb, pronoun, adverb, adjective etc. A POS tagger divides the feature space into smaller subsets: we can then use a language model to represent each word category separately. This classification of text helps us to understand which part-of-speech carries the maximum information with respect to importance and allows for reducing the vector space by discarding certain word categories.

We can modify the language model to only include words tagged with certain part-of-speech information (nouns, verbs, adjectives or combinations of them). The necessary information is gained by using the part-of-speech tags generated by the Hepple tagger [78]. Terms tagged as noun or proper noun are included in the language model for nouns, terms tagged as any kind of verb are used for building the feature set for the verb approach, and for adjectives the same.

Named Entities. We use named entity recognition (NER) to identify named entities in the news articles. The different named entity categories (locations, organizations, job titles, and persons) are then used as feature sets for the SVM. The first feature is the *Location* mentioned in the news article. A gazetteer list containing entries about locations like countries, cities, etc. is used to extract location information from the articles. The same approach is used to identify *Organizations* like "United Nations", and *Job Titles* like "President". Another feature set used consists of the *Persons* as found by the named entity transducer. These individual features are normalized by dividing the number of occurrences by the number of total occurrences of a category in each article.

Bigrams. To find out whether the co-occurrence of two consecutive terms in a text can help to assess the importance we investigated the usefulness of bigrams. Therefore we removed stopwords from the text and built bigrams out of the lemmas of all two consecutive terms. The immediate context of a term might be helpful to distinguish and estimate the importance. For example, the terms "bankrupt" and "country" might not indicate importance of a news article on a global scale, but the bigram "bankrupt_country" could indicate importance.

Topic Models. Supervised machine learning techniques, in particular support vector machines, need a set of features for each instance they are supposed to classify. For news prediction, extracting certain language features from news articles, such as term frequency counts, part-of-speech information, named entities, or bigrams, and using them as input for a SVM is one solution. We now present another solution based on LDA feature reduction.

In the most general form, we represent news with simple language models using term frequency vectors (TFV). For each news story, we use text from up to 7 different sources, and then combine the document as a TFV. This representation has a very large number of features and the data is very sparse. Using language models, we explore the effectiveness of SVM based importance classifiers on term frequency vectors. While this approach performs well, generalization is difficult due to the sparseness of features and redundancy. Thus, we propose the use of latent factors derived from dimensionality reduction of text as the features for a classifier. This dimensionality reduction not only generalizes and smoothens the noise but also decomposes the semantics of the text along different latent dimensions. To this end, we use latent Dirichlet allocation to perform the topic analysis.

An article about the 2008 Presidential elections in the US would then be represented as a mixture of the "election"-topic (say 40%) and an "Obama & McCain"-topic (maybe 30%) as well as some other latent topics (summing up to 30%). Using LDA, we learn a

Table 3.1: Number of occurrences for individual features in our corpus

Feature	**No. of Occurrences**
Nouns	39488
Verbs	3504
Adjectives	9794
Organizations	6560
Job Titles	589
Locations	3502
Persons	15543
Bigrams	726917

probability distribution over a fixed number of latent topics; for the purpose of classification, we treat the probabilities of latent topics as input features.

The weights of the support vectors of the support vector machine indicate which features separate the two classes best. In a classification setting with important news articles and unimportant news articles as training data, we can classify unlabeled new articles as important or unimportant. Further we can analyze the weights to identify the latent topics most indicative for the two classes.

To evaluate the effectiveness of different feature sets used to build predictive models we designed a component-based system which allows us to select the desired features sets for each run.

3.2.2 Evaluation

For evaluation we collected data from Google News [2]. We downloaded stories displayed in the "World"-category between Nov 15th, 2007 and July 3rd, 2008. This resulted in a first, small dataset consisting of 1295 topics, each containing between 3 and 5 articles from the time when the topic first appeared. For a second, larger dataset we collected 3202 stories from the Google news service and crawled 4-7 articles from different sources per story. These stories were collected over a period of one year (2008). We performed some cleaning of the HTML pages to the navigation and advertisement information on the pages as well as HTML tags. Stopwords were removed and the text was lemmatized. Table 3.1 gives an overview of the properties of our small dataset consisting of 5182 articles. The characteristics of the large dataset are similar.

Evaluating *importance* seems to be at the first glance a very subjective task. To ensure an objective measure and to clearly differentiate our work from what is known under "personalization" of news, we need a measure that is unbiased. In addition, this measure must provide the possibility of a fully automatic evaluation.

Because of the fuzzy definition of importance, we require robust importance feedback for training our classifier. We first considered manual annotations by human classifying articles as important or unimportant. Since this data is bound to have individual biases, we chose to use a statistic provided by Google News for each news item. This statistic is the *cluster size* reported for each article group (*topic*). Google News uses text clustering to group similar news; updates and growth in the news story is usually clustered in the same group. We argue that the relative importance of a news article or topic is dependent on cluster size; this concept is related to both popularity used by Social news sites, as well as citation analysis used for web ranking (e.g. Pagerank [141]). We also argue that cluster size is a robust indication of importance. Our task is thus to train classifiers and find which

[2] `http://news.google.com`

Table 3.2: Results for different thresholds (news cluster size) to separate unimportant and important topics based on language models

Threshold for	Number of News Topics		Correctly Classified	
Cluster Size	Unimportant	Important	Baseline	SVM (Nouns)
600	1062	233	82.01%	83.32%
500	1002	293	77.37%	79.46%
400	906	389	69.96%	71.97%
300	767	528	59.23%	65.71%
250	658	637	50.81%	66.26%
200	538	747	57.68%	65.41%

features are indicative of importance. Note that other sources can also be substituted for cluster size without any difference in the methodology.

For the language model approach, we represent each topic using a vector space model. If a topic consists of more than one article we use all terms and normalize the weights for each term by dividing it by the number of articles. The weights are standard tf*idf scores: $w = \mathrm{tf} * \log(\frac{N}{\mathrm{df}} + 1)$, with N = number of topics. Experiments with giving higher scores to terms if they appear early in the text, e.g. in the headline of an article or the first 3 sentences, gave worse results. Using a cut off value to remove low ranked terms from the topic vector yielded worse results as well.

For our experiments we have two different setups: The first one is a binary classification problem finding out which topics are important and which are not. Therefore we had to define the threshold for the cluster size to classify the topics. Table 3.2 shows the accuracy for different thresholds t using only nouns as features. Although our performance is only slightly better than that of the baseline[3] for higher thresholds, the recall for finding important topics is always higher (e.g. with a threshold of 500, recall for identifying the important topics is 35% with SVM (vs. 0% for the baseline) and precision is 58%). A manual evaluation showed that 500 is actually a good threshold for the binary classification in terms of what a human would consider important, thus 21% of the articles in the collection can be considered important. The following results are therefore based on a threshold of 500.

Secondly, we classified the data into different bins based on cluster size: we used one bin for the topics with cluster size between 0 and 500 ; one for 500 to 1000; 1000 to 2000; and one bin for the news topics with more than 2000 articles in cluster. The bins were assigned the values '1', '2', '3', and '4' respectively. Figure 3.1 shows the classification of the topics into the different bins.

Quality Measures. We measure the accuracy of classification by computing the 0–1 loss function for a test set. This function reports an error 0 if the correct label has been assigned, and 1 otherwise.

For two class problems, 0–1 errors are very indicative of accuracy: for multi-class problems however, this function would treat the misclassification of *highly important* as *unimportant* the same as *very important.* Clearly, the degree of misclassification should be considered as well. Therefore, we use *Mean Average Error* (MAE) and *Root Mean Square Error* (RMSE) using the label set $\{1, 2, 3, 4\}$.

$$MAE = \frac{1}{n} \sum (Label_{Predicted} - Label_{Actual}) \tag{3.4}$$

[3] Baseline: Classify all topics as unimportant for $t < 200$ and as important for $t >= 200$

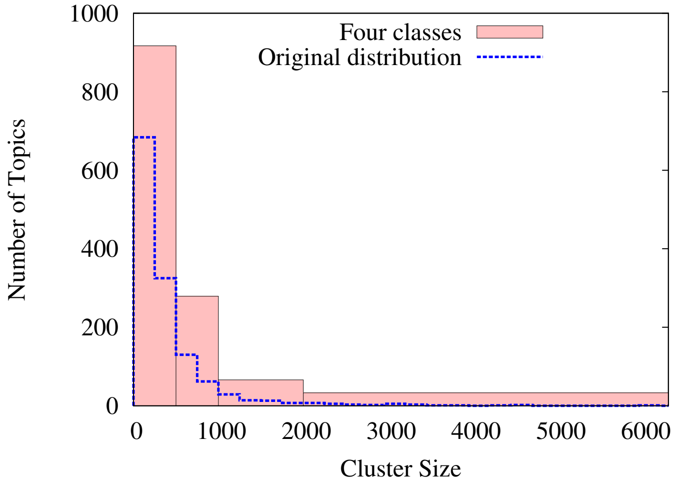


Figure 3.1: Distribution of the news article corpus into 4 different bins

$$RMSE = \sqrt{\frac{1}{n}\sum(Label_{Predicted} - Label_{Actual})^2} \quad (3.5)$$

In addition to reporting Accuracy and RMSE, we focus on Receiver Operator Characteristic — Area Under the Curve (ROC-AUC) values [149]. This ensures that we get comparable results even with different sized classes. For a two-class classification task, where we don't have an ordering or ranking of the results (e.g. a probability value that an instance belongs to one of the classes) the ROC-AUC value can be computed as: $\text{ROC-AUC} = \frac{1}{2}P_t \cdot P_f + (1 - P_f) \cdot P_t + \frac{1}{2}(1 - P_f)(1 - P_t)$ with P_f as the false positive rate and P_t the true positive rate.

Implementation. To pre-process the data and extract the different features we used GATE [42], an NLP-framework. For data mining and classification, we used the WEKA [201] toolkit. WEKA supports several standard data mining tasks, as well as data preparation steps. More specifically, algorithms for clustering, classification, regression, visualization, and feature selection are implemented. In addition, error evaluation protocols like cross-validation are supported. We used LingPipe's[4] implementation of LDA to extract the latent topics. All SVM results were obtained using 10-fold cross-validation and an RBF kernel.

3.2.3 Results

We first report on the results using the small dataset and analyze the different language features. Secondly we compare the results using language models with the results using topic models based on the larger dataset.

Prediction Accuracy. Table 3.3 summarizes the main results of our experiments; we notice that textual features are indicative of importance, but prediction accuracy is not very high. This trend does not change significantly as the ratio of training and test data is varied. We notice that binary prediction (2 classes: important and not important) is an easier problem, with up to nearly 80% accuracy achieved with linear SVMs. An important

[4] http://alias-i.com/lingpipe

Table 3.3: Results for binary classification with threshold 500

Used Features	**Binary Classification**		
	Correctly Classified	**Precision for unimp. Topics**	**Precision for imp. Topics**
Only nouns	**79.46%**	**82.9%**	57.6%
All features	79.31%	79.8%	**69.8%**
Bi-grams	78.92%	79.5%	67.2%
All named entities	78.53%	81.6%	55.0%
Verbs,nouns,adj	77.99%	81.7%	52.4%
Only persons	77.53%	81.2%	50.6%
Only job titles	76.37%	78.4%	40.8%
Only organizations	74.52%	79.9%	39.2%
Only locations	74.44%	81.2%	41.3%
Only verbs	71.89%	81.7%	37.7%
Only adjectives	71.00%	79.8%	34.2%

Table 3.4: Results for regression task using 4 classes: “extremely important”, “highly important”, “moderately important”, and “unimportant”

Used Features	**4 Class Regression Task**		
	Correlation Coefficient	**Mean Abs. Error**	**Root Mean Squared Error**
Only nouns	0.3535	0.5104	0.6771
All features	**0.4110**	0.5111	**0.6455**
Bi-grams	0.3985	0.5153	0.6507
All named entities	0.3453	0.5085	0.6751
Verbs,nouns,adj	0.3744	0.5034	0.6626
Only persons	0.3441	0.5032	0.6793
Only job titles	0.1421	**0.5020**	0.7863
Only organizations	0.1609	0.5707	0.7977
Only locations	0.1145	0.6312	0.9651
Only verbs	0.2087	0.6562	0.8782
Only adjectives	0.2434	0.5618	0.7452

observation we make is that our trained classifiers are highly accurate in predicting truly important news correctly. For detecting unimportant news, using only nouns yields the highest precision, whereas for predicting important topics, using all features results in more than 10% higher precision. However, several false positives are generated; this indicates that news reporting might be making some articles appear more important than they actual are.

Regression Accuracy. As explained earlier, a regression task is more sensitive to larger misclassification errors. Table 3.4 shows the experimental results for various strategies. Best regression results for the 4 class problem were achieved when using all features (highest correlation and lowest RMSE); however, the lowest MAE was observed when using only job titles. To verify this, we trained SVM models using a RBF kernel, as well as a C4.5 decision tree [152]. Linear SVMs were observed to provide the highest 4-class classification accuracy.

Most Discriminative Features. We mined the SVM model to find the most discriminative features; the features with the highest weights are indicated in Table 3.5. We note

Table 3.5: Three most indicative terms for each feature for binary classification. DF is the document frequency with a total of 5182 articles (1295 topics) in the corpus

Category	**Weight**	**DF**	**Label**
Person	0.018	58	Mohammed
Adjective	0.018	86	state-run
Verb	0.016	425	head
Adjective	0.016	535	dead
Bigram	0.016	17	dead police
Verb	0.015	295	gather
Adjective	0.014	57	authorized
Bigram	0.014	29	close ally
Bigram	0.013	151	identity trap
Noun	0.013	890	violence
Noun	0.013	741	car
Job Title	0.013	1199	spokesman
Person	0.013	53	President Vladimir Putin
Location	0.012	18	eastern Baghdad
Noun	0.012	188	reaction
Person	0.011	115	Prime Minister Brown
Job Title	0.011	1069	official
Job Title	0.011	43	CEO
Location	0.010	105	Bali
Location	0.010	119	Scotland
Verb	0.009	1008	remain
Organization	0.008	22	Labor Party
Organization	0.008	106	Senate
Organization	0.007	117	Pentagon

that world leaders are identified as influential features (Puting, Brown), and crime related events (e.g. dead police, violence) also figure highly. News containing terms like ”close ally”, “state-run” (related to Political news) is also considered more important than others. Organizations, locations, and job titles usually carry smaller weightage than nouns, verbs, and adjectives; this trend is noticeable in both 2-class and 4-class classification.

Changing Importance of Terms. We also investigated if the weights of features change with time; this seems intuitive as really important world events can make a location or a person suddenly famous. In Figures 3.2, 3.3, 3.4 and 3.5, we see the importance the classifier assigned to selected terms over a period of six months. We trained six classifiers, one for each month to compare the weights over time. We notice exploding importance of certain locations for different months (e.g. “Northern Ireland” in March, “Palestine” in June), vs. rather steady weights for general terms like “warfare” or “bomb”. The changing political environment also influences the importance of politicians’ names,“Musharraf” is steadily loosing importance, whereas “Abbas” experiences an importance peak in the month of May. Terms like “violence” and “arrest” are also likely to suddenly become more important. These interesting trends are prominent in our data collection; we expect that analysis over longer term data will lead to a robust set of features and higher accuracy.

Example Predictions. Table 3.6 gives an overview of the top 10 news stories as predicted by our algorithm. Also the 10 most unimportant ones according to our predictions are shown. Although the actual ranking is debatable, the coarse classification is rather convincing, giving relevant global news (“5.4 Million Dead in DRC Since 1998, Says New Survey”)

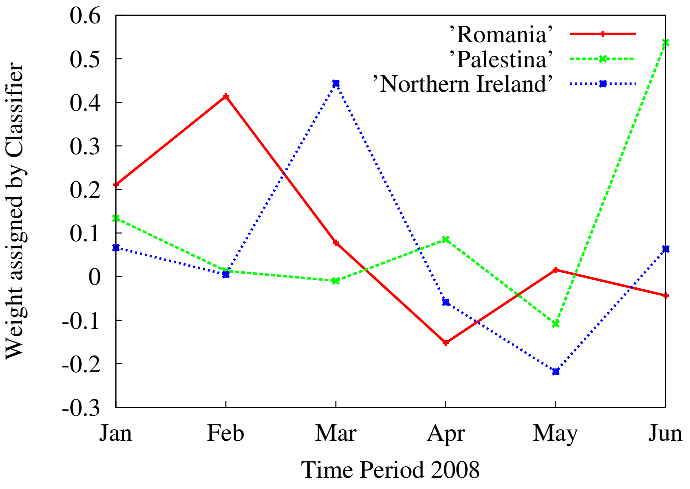


Figure 3.2: Assigned weights over different periods for selected locations

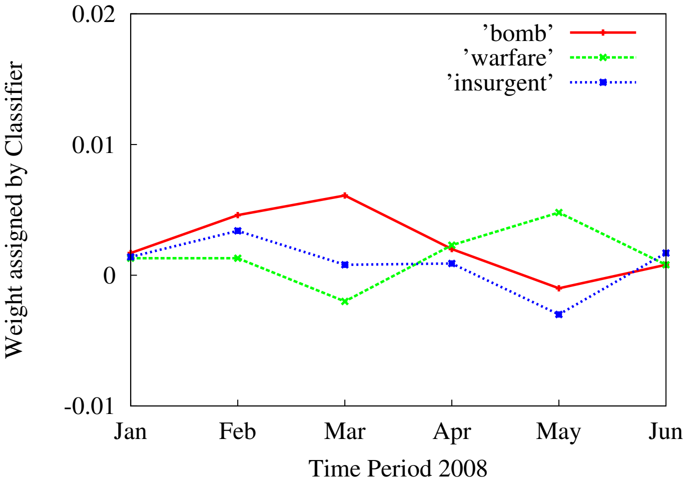


Figure 3.3: Assigned weights over different periods for selected nouns

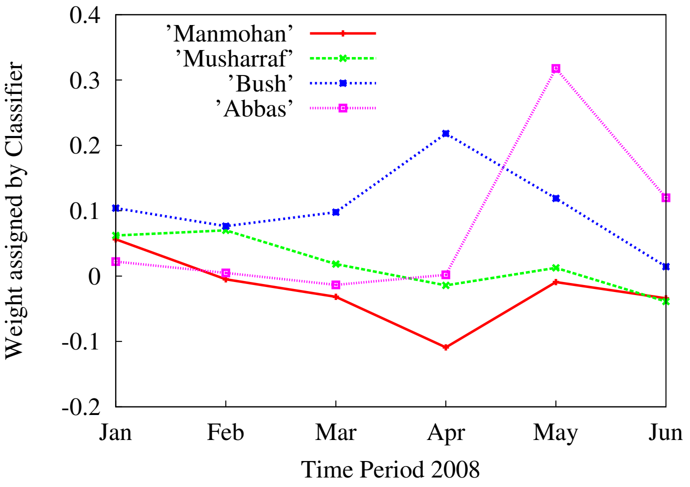


Figure 3.4: Assigned weights over different periods for selected persons

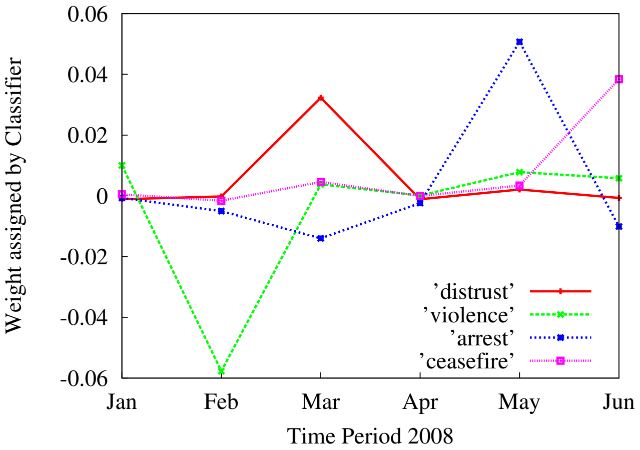


Figure 3.5: Assigned weights over different periods for selected nouns

more credit than local news which are rather interesting and important for a small group of people ("New faces for Irish politics").

Topic Models. To compare the LDA approach to the language model approaches for importance prediction, we also applied the described methods to the large dataset. Namely part-of-speech tagging and named entity recognition were used to get an enhanced language model representation for each news story. With this method, we get ROC-AUC values of up to 0.683 when doing a binary classification based on two equally sized bins (threshold ∼250) compared to 0.728 with our LDA approach. This is an increase of more than 10%.

Table 3.6: Top 10 and flop 10 stories as predicted by our system

Rank	Top Stories	Flop Stories
1	5.4 Million Dead in DRC Since 1998, Says New Survey (01.22.08)	New faces for Irish politics (04.14.08)
2	Bush and Sarkozy seek united front against Iran (07.14.08)	Australia to remove almost 100 anti-gay laws (04.30.08)
3	Myanmar death toll soars (05.17.08)	Bangladesh government announces national elections (05.12.08)
4	Serb protesters attack U.S. Embassy (02.21.08)	Motive still unknown as serial killer faces rest of life in prison (02.22.08)
5	Bomb Blast in Yemen Kills 18 at Mosque (05.02.08)	BAE chief subpoenaed in U.S. over Saudi arms deal (05.18.08)
6	Democrats take White House campaign to Puerto Rico (05.24.08)	As Bush leaves Mideast, he gives Arab leaders a to-do list for reform (05.19.08)
7	Raj Thackeray arrested in Mumbai, gets bail (02.12.08)	Teachers struggle with immigrant pupil influx (03.21.08)
8	MPs back animal-human embryos for research (05.19.08)	Rudd targets UN council seat (03.30.08)
9	Mugabe Is Sworn In to Sixth Term After Victory in One-Candidate Runoff (06.30.08)	Iraqi militia to hear Saturday whether to resume fighting (02.20.08)
10	Iran Threatens To "Explode" Ships (01.08.08)	No More Deja Vu: A Tenacious Negotiator Cuts A Deal On Hebron (02.27.08)

Table 3.7: Results for different language features and binary classification on equally sized bins

Feature	Accuracy	ROC-AUC
All Types	64.52%	0.683
Verbs, Nouns, Adj.	63.65%	0.677
Nouns	62.90%	0.668
Named Entities	61.84%	0.645
Verbs	58.68%	0.606
Adjectives	58.34%	0.608
Persons	57.78%	0.598
Locations	56.62%	0.589
Jobtitles	55.56%	0.581
Organizations	55.56%	0.573

Table 3.8: Results for different number of latent topics and binary classification on equally sized bins

No. of LDA Topics T	Accuracy	ROC-AUC
50	63.59%	0.682
100	65.58%	0.716
250	64.83%	0.709
500	65.99%	0.720
750	65.15%	0.717
1000	66.27%	0.728
2500	65.87%	0.723

Two-Class Classification. In Table 3.7 the results for filtering the input data making use of part-of-speech information and Named Entity Recognition is shown. The numbers indicate that the pre-selection of certain word types is decreasing the prediction accuracy. Overall best accuracy is 64.52% achieved using all word types. Nouns tend to have a higher predictive value as e.g. persons. In the following we will compare our results only with the language model approach keeping all word types, since this yielded the best results.

Using LDA to reduce the number of features improves not only efficiency and interpretability but also accuracy. We evaluated the performance of our algorithm varying the number of LDA topics generated from the news data. The ROC-AUC values are between 0.682 for 50 LDA topics and 0.728 for 1000 (see Table 3.8). The best accuracy is 66.27%. The higher the number of latent topics, the more specific the LDA topics.

Regression Results. The correlation coefficient is 0.47 for using LDA compared to 0.39 when using the language model approach. Root relative squared error is with 89.14% rather high but still better then using BOW. Figure 3.6 shows the ROC curves for varying the threshold. We therefore do a normal regression and then systematically lower the threshold for a story to be important starting from 1.0. For each threshold we get a false positive rate and a true positive rate. The blue line ($f(x) = x$) indicates a random algorithm. Our results are significantly better for both, BOW (green line) and LDA (red line).

LDA Topics. Table 3.9 shows the three top ranked LDA topics with respect to information gain. Analysing the model built by the classifier reveals that topic 128 is indicative for unimportant news articles whereas the other two indicate important news. Since we did this evaluation using 250 LDA topics to represent our documents, some LDA topics

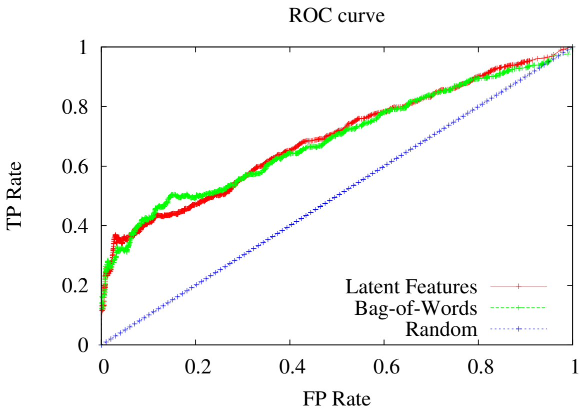


Figure 3.6: ROC curves for different thresholds (cluster size) to separate unimportant and important topics in the regression setup using latent features and bag-of-words

contain actually two “topics” (oil, nigeria, indonesia). Also note that the term “government” appears in two topics with different probabilities. Other LDA topics indicating importance are e.g.: “McCain, Obama, campaign” or “gas, Ukraine, Russia”.

Table 3.9: Top features based on information gain. First two indicating important news; third one indicating unimportance. For each word also the number of occurrences in the corpus is displayed, as well as the probability that the word belongs to the topic.

Topic84			**Topic197**			**Topic128**		
Word	**Count**	**Prob**	**Word**	**Count**	**Prob**	**Word**	**Count**	**Prob**
afghanistan	5223	0,120	oil	2256	0,135	able	1803	0,069
afghan	1801	0,041	nigeria	363	0,022	browser	1786	0,068
nato	1732	0,040	company	334	0,020	content	1298	0,049
taliban	1288	0,030	militant	302	0,018	view	1275	0,049
troops	1153	0,027	barrel	280	0,017	style	1206	0,046
country	869	0,020	production	263	0,016	enable	1200	0,046
kabul	709	0,016	crude	246	0,015	sheet	1187	0,045
force	665	0,015	niger delta	240	0,014	css	1136	0,043
security	573	0,013	attack	225	0,013	bbc	1017	0,039
fight	569	0,013	pipeline	211	0,013	internet	720	0,027
karzaus	562	0,013	field	198	0,012	full	664	0,025
government	552	0,013	region	197	0,012	responsible	645	0,025
military	537	0,012	day	190	0,011	upgrade	644	0,025
pakistan	490	0,011	nigerian	190	0,011	consider	632	0,024
mission	447	0,010	group	183	0,011	external	627	0,024
ally	443	0,010	government	182	0,011	current	621	0,024
send	435	0,010	mend	170	0,010	experience	615	0,023
war	366	0,008	gas	153	0,009	software	611	0,023
us	345	0,008	major	117	0,007	best	599	0,023
president	344	0,008	industry	114	0,007	visual	578	0,022

3.3 Related Work

Automatic Ranking of news is a rather recent discipline. There is however a long tradition trying to explain the importance of some events or news. This research is traditionally conducted by communication theorists or journalists. Starting with Lippmann [121] who introduced *news value* as a measure of importance followed later by Østgaard [140] who researched which *factors* make a piece of news important. Kepplinger and Ehmig [98] tried to predict importance (*newsworthiness*) of news based on a manual content analysis of experts. With the access to a lot of news online and the overload of a single user, automatic ranking or filtering of news becomes very important.

Most commercial news sites have some mechanism to rank different news articles. While some providers rely on expert editors, other sites like Digg[5] and Slashdot[6] rely on social human filtering. Google News provides an aggregate news service which is automated, and close to our objective. Although they are not publicly available, some of the features they take into account can be inferred from analyzing the pages. Das et al. [44] disclose the use of large scale collaborative filtering and text clustering as important constituents of the Google news algorithm. A major part of the Google news approach is the identification of a topic to group articles dealing with the same event together. Classification of the news articles into predefined categories precedes this clustering. Obvious features to rank these different topics within one category are the size of the cluster, the time the articles were published, and the sources who published the articles. These features are also relevant for ranking different news articles within one topic.

Del Corso et al. [41] describe a ranking algorithm for news sources and articles. They take into account that an important news topic generates many articles from different sources. They also included a mechanism to mutually reinforce scores of articles and their sources. Because they are looking at a stream of news, time awareness of the algorithm is a crucial point, since old news are considered less valuable than new news. A last feature of their ranking algorithm is the possibility of online processing of the data and of ranking the different articles on the fly.

A slightly different approach is presented in Yao et al. [206]. In their paper, the authors make two assumptions on what is an important news article: First, important news have a prominent spot in news Web pages; second, important news are covered by various news sites. This allows an internal ranking for each article on news sites based on visual layout and mirrors the relation between an event and articles about this event. These relationships between Web pages and news, as well as between events and news are used to model a news-importance relationship by a tripartite graph where Web pages, news, and events are nodes. Each homepage gets a weight according to its credibility; each news article and each event get weights according to their importance. The weights are computed using an iterative algorithm exploiting the graph information. The evaluation shows that results improve significantly by taking into account both measures: visual layout and event clustering information. Notice that this approach is not suitable for news feeds (e.g.) based on RSS.

This approach is modified by Hu et al. by [86] by changing the graph structure and only considering news and sources and the corresponding relations between them. The use of a semi-supervised learning algorithm is proposed to predict the recommendation strength of a news site for articles on other news sites, which leads to more edges in the graph and yields a better performance for the algorithm. Similarity between articles is

[5] http://digg.com

[6] http://slashdot.org

measured using a vector space model and the relation between sources and articles are weighted using visual layout information.

All these systems have one common feature; they use information from news pages on the Internet, either taking the number of similar news articles into account or the internal ranking of articles within news pages. The drawback of these approaches is that they give an overview of what news are there and they rank these news items without regarding their intrinsic important-importance. However, newspapers and news sites have to publish articles even if nothing really important happened; thus all news which is on the front page of a news site, is not equally important. The result of the described systems is always relative for a given time period. Further, there is an implied dependence on social feedback, or duplication; however, this information is not necessary available when a news item is reported.

Content based prediction is thus much more useful, since it can immediately classify news; clearly, content analysis is far more difficult, thus requiring a better understanding of the news domain. Our proposed approach explores a feature-space which is available at publish time, and hence does not suffer from latency issues. Further we try to produce absolute numbers, allowing the user to set a threshold, which marks the boundary for new news articles to be presented to the user.

Our Contributions. We have explored the use of textual features and latent features for creating an understandable prediction model for news importance. Experimental results indicate that this is a hard problem, with pure textual information being insufficient for creating accurate classifiers. We report interesting observations of nouns being more indicative of importance, and the sensitivity of classifiers to world events. The main goal of using topic models was to make the importance prediction more accessible in the sense of easy interpretable results. We have shown that LDA can achieve this. It is possible to identify general news events, e.g. the war in Afghanistan, and predict the importance of future articles dealing with this topic. Also general events like elections were identified by LDA and help to predict the coverage of other future elections in the media. In conclusion, we explored a new approach to the problem of finding important news as a classification problem using LDA topic mixtures. The results show that accuracy using LDA is higher than using language models. We can find important articles with an accuracy considerably better than random, and around 10% better than previous approaches.

CHAPTER

FOUR

SENTIMENT CLASSIFICATION

In this chapter we present an approach for automatically assigning sentiment values to terms based on their context. We therefore employ topic modeling to identify the topical context and generate sentiment lexica for these topics.

4.1 Introduction

The rapidly increasing popularity of user-generated content is based on the availability of suitable and easy to use mechanisms for publishing blog articles, product reviews, comments on news events, and contributions to discussion forums. The blogosphere has attracted an active web community in the recent years, and has become a popular environment for sharing experiences, opinions, and thoughts on a variety of issues. Topics discussed range from rather casual themes such as sports, concerts, and celebrities to more complex and polarizing political ones such as abortion, elections, and immigration. Large online-review communities on platforms such as Epinions, Amazon, or IMDB contain a variety of opinionated views on books, movies, and consumer electronics. Content sharing platforms such as YouTube and Flickr provide different social tools for community interaction, including the possibility to comment on existing resources.

Techniques for automatic extraction of sentiments and opinions allow for a variety of applications such as opinion-oriented search, prediction of trends, summarization of product aspects, and filtering of flames in newsgroups. Liu et al. [122] make use of sentiment analysis techniques to predict movie incomes by mining blogs, carrying the potential to utilization in market analysis and business planning. Lu et al. [125] use short comments on products to provide an aggregated view on user opinions about thematic aspects such as “shipping”, “communication”, or “service”. Turney and Litman [183] mention several additional existing

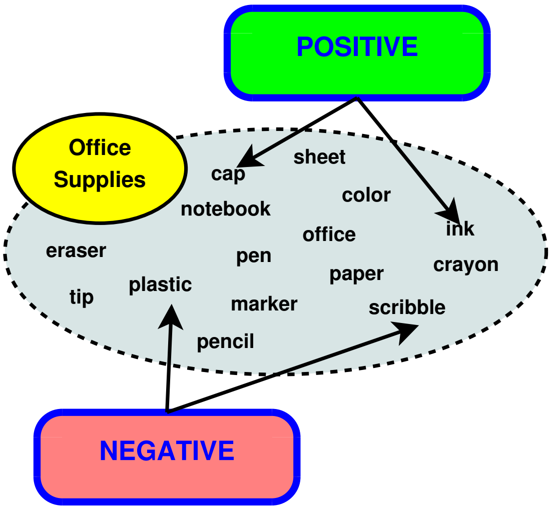


Figure 4.1: An Example of a Topic with Topic-Specific Sentiment Assignments

and potential applications of sentiment analysis such as corpus linguistics, aggregated views in the form of sentiment timelines, and even AI components in computer games responding in a more realistic way to the player's textual input.

Many of the above mentioned applications build on *sentiment lexica* which provide information on the typical polarity of words. For instance, SentiWordNet [8], a lexical resource built on top of WordNet [58], assigns triples of *sentivalues* (corresponding to positive, negative, and neutral sentiment of a word). Turney and Litman [183] use a small seed set of polarized terms for automatically extracting additional sentiment terms from larger text corpora.

Although there are general terms that almost always carry the same sentiment, their polarity can be highly *context-dependent*, as pointed out, for instance, by Nowson [137] ("Scary Films Good, Scary Flights Bad"). This makes sentiment analysis across different domains and contexts a challenging task. In the last years some attempts have been made towards creating context-dependent sentiment lexica. Kanayama and Nasukawa [97] separately created lexica on data from four discussion boards on different topics which were considered as the domains. A rather fine grained lexicon generation approach is described by Bross and Ehrig [28] where combinations of context-dependent polarity terms and entities are extracted (e.g. "long battery life" vs. "long delivery time").

Figure 4.1 shows example entries from our lexicon. The top-15 terms composing the latent topic "office supply" are shown. Within this topical context, our algorithm identified a negative sentiment value attached with "plastic" and "scribble", whereas "cap" and "ink" are perceived as mainly positive. Especially the sentiment value of "plastic" is highly topic dependent, since for other products plastic is associated with a rather positive sentiment value due to its light weight and inexpensive production.

The work on sentiment lexicon generation presented here is placed in between lexica in broad domains [97] and fine grained lexica on specific opinion-entity pairs [28]. We aim to determine the polarity of words in a *topical context*. To this end, we make use of a large corpus of review texts as well as the accompanying star ratings found in many Web 2.0 applications to automatically extract the sentiment of a term in a given context. Our approach combines topic extraction using latent Dirichlet allocation with discriminative feature extraction techniques based on information theoretic paradigms for exploiting *user ratings*. In addition, we assign sentiment values not only to single terms but also to phrases and named entities represented by different-sized n-grams. Experiments for finding topics and associate sentiment values to terms were conducted on a dataset containing $27,375$ Epinions reviews. We performed a classification-based evaluation on Epinions as well as on a well-known multi-domain sentiment dataset [26]. In addition, we conducted a direct manual evaluation of sentiment lexicon entries.

Problem Statement and Description. Individual terms or phrases can carry sentiment. The goal of a sentiment lexicon is to assign each term to a sentiment score reflecting its positivity or negativity. Optionally, a value for objectivity can be assigned to indicate how subjective a term is. In its simplest form, sentiment scores s_{t_i} are assigned to terms t_i to form an entry in a sentiment lexicon: (t_i, s_{t_i}), with $t_i \in V = \{t_i, \ldots, t_n\}$ and $s_{t_i} \in [-1.0, +1.0]$ to indicate negative (-1.0) or positive $(+1.0)$ sentiment or 0.0 for neutral sentiment.

Since our goal is a context-dependent lexicon, we additionally need to consider the context of a term. This context can be defined by the surrounding terms, the containing sentence, or paragraph. Also the whole collection or single documents can be considered the context for a certain term. For example in “cold beer”, cold has a positive sentiment whereas in “cold coffee” cold is something negative.

4.2 Topic Models for Generating Context-Dependent Sentiment Lexica

In this section we describe the construction of a context-dependent sentiment lexicon. Topic models serve as the decisive context and user-assigned star ratings from a review platform are exploited to identify the polarity of terms using an adapted mutual information measure.

4.2.1 Approach

The goal of our context-dependent sentiment lexicon is to assign sentiment values to terms depending on their topical context. This context can be defined by the surrounding terms [28], the comprising sentence [195], paragraph, document, or a whole domain [64]. In our approach, we propose to consider the document as the decisive level of granularity. Within a corpus of user-generated reviews, this allows for identifying different product categories (or services) and computing sentiment lexica for them or their aspects. How fine-grained a context is defined depends on the number of topics for the whole corpus, and is set as a parameter for the topic detection algorithm.

Depending on the document context, sentiment values can significantly differ for individual terms. If, for instance, the term “rail” occurs in a hotel review, it is mostly associated with a negative sentiment due to the noise of the trains. On the other hand, in a football stadium review, “rail” will be more likely associated with a positive sentiment because of improved accessibility. In order to find the positive and negative terms within a context or product category, we conduct a discriminative analysis of the terms in the reviews that exploits user-assigned star ratings.

Preprocessing. In order to preprocess the review data, we first applied the Stanford part-of-speech tagger[1] and kept only nouns, verbs, adjectives, and adverbs. We then used WordNet[2] to find the lemmas of each term. After that, we generated a list of n-grams from each review; in our experiments we obtained the best results for a combination of unigrams, bigrams, and trigrams. Finally, we discarded all n-grams occurring less then 5 times in our corpus in order to eliminate idiosyncratic terms. We also experimented with the removal of stopwords using stopword lists or term frequencies within the corpus. However, the best results were achieved by removing only the verbs “to be” and “to have”. After these steps, each review was represented as a list of POS-tagged, lemmatized n-grams.

Automatic Topic Identification. Instead of considering explicitly given categories, we automatically extract topics using latent topic analysis. Even in the special case of reviews, where each review is – often in a hierarchical manner – categorized into exactly one product group, automatic topic identification can be beneficial. For instance, for books, the category “Media” might not appropriately reflect the topics for our sentiment lexicon, as the category is possibly too broad. Since our objective is to provide topic-specific lexica for general purposes, we aim to detect the domain of a text automatically. For this step, we employ latent Dirichlet allocation (see Section 2.3.2), which additionally allows for a probabilistic assignment of different topics to a single review.

By applying LDA, we are able to represent latent topics as a list of n-grams with a probability for each n-gram indicating the membership degree for the topic. Furthermore,

[1] http://nlp.stanford.edu/software/tagger.shtml

[2] http://wordnet.princeton.edu/

for each product review in our corpus we can determine through topic probabilities $P(z_j \mid d_i)$ to which topics it belongs and to which degree.

In the next step, we assign the documents in the corpus to latent topics. To this end, for each latent topic, we iterate over the reviews and assign all reviews to this topic based on the topic probability. Thus, our corpus is divided into overlapping sub-corpora with each one representing one latent topic. Based on the document collections for each topic, we generate the topic-dependent lexica in the next step.

Lexicon Construction. In order to build topic-dependent lexica, we employ statistical methods for analyzing a large amount of product reviews covering a variety of products and services. Our hypothesis is that product reviews together with their star rating can be used to identify terms with positive or negative semantic orientation. A user assigning the maximum number of stars to a product will be likely to write a positive review using positive terms to describe the product. Conversely, a low rating is likely to be reflected by usage of rather negative terms. In general, we can consider two training corpora for each topic: One containing all positively rated documents and one containing all negatively rated documents. We further make use of the probabilities assigned to terms and documents through LDA.

The LDA-based approach described in the previous subsection generates latent topics for the whole corpus. Each document d_i is modeled as a distribution over topics Z, and can be represented as a mixture of topics as follows:

$$\sum_{z_j \in Z} P(z_j \mid d_i) = 1 \tag{4.1}$$

Reciprocally, each topic z_j is generated by different documents. By defining a threshold on the topic probability $P(z_j \mid d_i)$ for each document d_i, we can employ these topic generating documents to obtain a topic-specific corpus of reviews. The probability $P(z_j \mid d_i)$ is used as a weighting factor for each document d_i.

For each latent topic, we now have a corpus with weighted documents and assigned star ratings. To compute sentiment values for a latent topic, we assign documents to the "positive" or "negative" class $C \in \{pos, neg\}$ with a certain probability depending on the assigned star ratings (class probability). In order to identify the most discriminative terms, we adapt the mutual information measure. Pointwise mutual information (PMI) is an information theoretic measure to compute the mutual dependence between two random variables. Mutual information has been used in the past to do feature selection for text categorization [204, 211] and for exploiting the occurrence of emoticons in blogs to build an emotion lexicon [203]. In our case, we are interested in the dependence between the occurrence of an n-gram w and the membership to the positive or negative class $c \in \{pos, neg\}$. In its general form, Pointwise mutual information is defined as

$$PMI(w,c) = \log\left(\frac{P(w,c)}{P(w) \cdot P(c)}\right) \tag{4.2}$$

where w is a n-gram, c is a class, and $P(w,c)$ is the probability that n-gram w occurs in a document of class c.

In order to account for different degrees of topic membership and rating polarity, we incorporate the topic probability of a document and its rating class probability into the general pointwise mutual information computation (Eq. 4.2). Instead of a hard rating class assignment, we assign a probability to indicate the class membership of a document based on its star rating x using a sigmoid function $f(x)$:

$$f(x) = \frac{1}{1 + a^{b-x}} \tag{4.3}$$

with a defining how strong the positive and negative star ratings should be discriminated, and b defining the neutral star rating. In our experiments, we used a setting of $a = 4$ and $b = 3$. We adapt pointwise mutual information as follows to incorporate the topic and rating probabilities:

$$\begin{gathered}\mathrm{PMI}_{ad}(w, c, z) = \\ \log\left(\frac{\frac{1}{|D|}\sum\limits_{d \in D} P(w \mid d) P(c \mid d) P(z \mid d)}{P(w) \cdot P(c) \cdot P(z)}\right)\end{gathered} \tag{4.4}$$

where $|D|$ is the total number of documents, and $P(w \mid d)$ is computed using tf-idf. $P(c \mid d)$ is estimated based on the assigned star rating using the mapping function $f(x)$ defined in Equation 4.3. Thus, $P(c \mid d) = f(x)$ for $c = pos$ and $P(c \mid d) = 1 - f(x)$ for $c = neg$.

For computing a sentiment score for an n-gram w in a given context, we compute $\mathrm{PMI}_{adapt}(w, pos, z)$ for the positive class and $\mathrm{PMI}_{adapt}(w, neg, z)$ for the negative class. The context-dependent sentiment value (CDSV) is then computed as

$$\mathrm{CDSV}(w, z) = \mathrm{PMI}_{ad}(w, pos, z) - \mathrm{PMI}_{ad}(w, neg, z) \tag{4.5}$$

with $\mathrm{CDSV}(w, z) = 0$ indicating that, in the context of topic z, term w is neutral with respect to sentiment, and negative/positive CDSV scores indicating negative/positive sentiment. For each latent topic. we can compute the CDSV for each term and accordingly identify the discriminative n-grams between the positive and the negative ratings. In addition to the individual sentiment lexica for each topic, we also computed a general, topic-independent sentiment lexicon using pointwise mutual information on the whole corpus to compare the results with the topic-dependent approach.

In our experiments, we aim to infer the polarity of a term in a review by considering the review text as context. To this end, we compute the context-dependent sentiment value for a term w within a document d as:

$$\mathrm{CDSV}(w, d) \quad = \quad \sum_{i=1}^{|Z|} p(z_i|d)\ \mathrm{CDSV}(w, z) \tag{4.6}$$

Optionally, introducing additional weights, combining the score with a context-independent component, or normalizing the values could be beneficial depending on the application area of the lexicon.

4.2.2 Evaluation

In this section, we present the results of our evaluation for automatic generation of a context-dependent sentiment lexicon. First, we describe our strategy for gathering a corpus of reviews from the rating platform Epinions, and elaborate on the characteristics of our dataset. Then, we present the outcome of our evaluation methodology: First, we show the outcome of a user evaluation for manually assessing the context-dependent lexica. Second, we compare the results of different methods for predicting the star rating of reviews. This problem is known in the literature as sentiment classification. Systems which tried to solve it usually include either machine learning [144], sentiment lexica [182], or both [5, 43]. This makes it an adequate testing environment to compare the performance of our lexica with state-of-the-art lexicon generation methods.

Data. For our experiments, we employed a large dataset crawled from Epinions[3] in 2010. Epinions is a rating and review platform for a variety of different products and services, ranging from cars to football stadiums, hotels, and DVDs. Users can rate products on a five star scale and write a review about their experience to justify the given rating. The products are classified into 16 distinct categories with 318 sub-categories. To assess the influence of the size of the training data on the performance of our algorithm we generated two different sized datasets: A *small* one containing 7,500 reviews from 15 of the categories with 100 reviews for each star rating and category and a *large* one with 27,375 reviews from the 11 categories containing approx. 500 reviews for each star rating and category.

In order to compare our results with previous work, we additionally used a test set designed by Blitzer et. al. [26] to evaluate domain adaption for sentiment classification. It consists of product reviews from Amazon[4] for four different product categories: books, DVDs, electronics, and kitchen appliances. Each review has an associated star rating with reviews having a rating > 3 labeled positive and < 3 labeled negative. Reviews with a rating of 3 were discarded because their polarity was considered ambiguous. This led to 1,000 positive and 1,000 negative reviews for each category.

User Evaluation. We first conducted a manual evaluation to directly assess the quality of the sentiment values for the latent topics. In order to provide a topic context, the sentiment terms were presented to the assessors along with a latent topic. A topic was represented by the top-20 terms with the highest probability of topic membership. For each topic, a positive, a negative, and a neutral term associated with the topic were shown in a random order. Ten assessors were asked to order them from positive to negative. An example from the manual test set with high inter-rater agreement is shown in Table 4.1. The topical context is given on the left by the most likely terms for the topic. Within this "hotel" context, most judges considered the term "spa" as the most positive one, "motel" as the most negative one, and "morning" as rather neutral, which is in conformance with our algorithm. Users associate luxury and recreation with the term "spa" leading to a positive attitude, whereas "motel" is associated with low budget accommodation and rather negative feelings.

We evaluated all of the 75 contexts from the lexicon computed using 75 latent topics. No context-independent component was added to the sentiment values, making the ordering a challenging task. After each assessor ordered the triples according to their sentiment value, we analyzed the inter-rater agreement and the correlation with the automatically assigned sentiment values generated by our algorithm. In order to measure the inter-rater agreement, we used Kendall's W, which can take values between 0.0 and 1.0, with 1.0 indicating perfect agreement. For the correlation of the automatically generated sentiment order, we compared the ranking assigned by the user with the one obtained through the algorithm. To this end, we used Spearman's rank correlation coefficient ρ, whose values

[3] `www.epinions.com`

[4] `www.amazon.com`

Table 4.1: Examples from the user evaluation: The topical context is shown on the left; on the right, three terms are shown along with the user assigned ordering from negative to positive.

Topic 11
hotel stay bed night
pool desk bathroom service
floor door restaurant staff
breakfast inn lobby guest
location casino bar check

Terms	**Order**
morning	2
motel	1
spa	3

range from -1 for perfect negative correlation to $+1$ for perfect correlation; a value of 0 indicates no correlation.

Review Classification. In order to evaluate our approach in a large-scale and automatic way, we conducted additional experiments on using our lexicon to classify reviews. To this end, we tested the effectiveness of different methods to predict the star rating associated with a review. This is a typical application scenario for sentiment lexica. Positive words in a review indicate in most cases a positive rating for the discussed product whereas negative words indicate that the author of the review was not satisfied with the product. This classification approach is also popular to classify blog entries in the context of sentiment analysis [66]. Our ground truth for classification is the star rating assigned by a user to a product, and the input data consists of the review written by this user. The algorithms should rank the given test reviews from negative to positive, and approximate the ground truth ranking as close as possible.

In our experiments, we used 5-fold cross-validation with 20% test data and 80% training data based on the Epinions datasets described in Section 4.2.2. In addition, we evaluated our algorithm on the test set described in [26]. We compare the results from our context-dependent sentiment lexicon (CDSL) with the results from two static, domain-independent lexica: SentiWordNet [8] and MPQA [195], as well as with domain specific lexica based on pointwise mutual information using a seed set of terms [183].

Computing Scores for Baselines. For the static lexica, computing scores for reviews was done in a straightforward manner: For each word in the review, we looked up the sentiment score in the lexicon and computed an average over all words. This method turned out to achieve better results than counting positive and negative terms in the review using a threshold on the sentiment scores in the lexicon as proposed e.g. in [53]. For SentiWordNet we also averaged over different WordNet synsets if a term had more than one sense. An example for WordNet synsets and associated SentiWordNet scores is shown in Table 4.2. For comparing scores, we also computed sentiment values for bi-grams, tri-grams, etc. by averaging over the single terms of the n-gram. For the MPQA lexicon we assigned a score of 1.0 to all positive terms labeled “strong subjective” and 0.75 to the positive terms labeled “weak subjective” (-1.0 and -0.75 for negative terms respectively). These static, context-independent lexica result in a positive or negative average score for each review that is then used to predict the star rating assigned by the user.

The domain-dependent baseline is an implementation of the approach described in [183]. For each term t in the review to be classified we computed a sentiment score as:

$$\text{Score-PMI}(t) = \sum_{\text{p}\in\text{POS}} \text{PMI}(t,p) - \sum_{\text{n}\in\text{NEG}} \text{PMI}(t,n) \tag{4.7}$$

Table 4.2: An Example of SentiWordNet for the adjective “cruel”: The different WordNet senses (synsets) are represented by the terms with a sense number in the synset

WordNet Synsets	**SentiWordNet Scores**		
	Positive	**Negative**	**Objective**
cruel(adj,3) brutal(adj,1)	0.0	0.625	0.375
harsh(adj,5) brutal(adj,3) cruel(adj,4) rigorous(adj,3) unkind(adj,4)	0.0	0.625	0.375
unkind(adj,3) cruel(adj,1)	0.5	0.125	0.375
fell(adj,1) brutal(adj,2) cruel(adj,2) savage(adj,1) barbarous(adj,1) roughshod(adj,2) vicious(adj,1)	0.0	0.625	0.375
Average	0.125	0.5	0.375

where POS is a list of positive seed words and NEG a list with negative ones. Pointwise mutual information (PMI) between the terms is computed as:

$$\mathrm{PMI}(\mathrm{term}_1, \mathrm{term}_2) = \log_2 \left(\frac{P(\mathrm{term}_1, \mathrm{term}_2)}{P(\mathrm{term}_1) \cdot P(\mathrm{term}_2)} \right) \tag{4.8}$$

where $p(\mathrm{term})$ is the probability of a term appearing in a sentence, and $p(\mathrm{term}_1 \& \mathrm{term}_2)$ denotes the probability that term_1 appears in a sentence together with term_2. We also experimented with computing the co-occurrence on a document level but results on sentence level proved to be superior. We discarded the original seed sets that consisted of only 7 positive and 7 negative terms in favor of a larger seed set of around 1,000 positive and 1,000 negative adjectives and adverbs described in [193]. The Epinions datasets were used to compute the probabilities for each latent topic as described in Section 4.2.1. The optimal threshold for considering a review positive or negative was determined using cross-validation since the “natural” threshold of 0.0 would have classified most reviews as being positive.

Computing Scores for CDSL. Generating the topic-specific lexica as described in Section 4.2.1 provided us with a latent topic model of the data and with a sentiment lexicon for each latent topic. To compute a cumulated sentiment score for each review in the test set, we had to combine different topic-specific lexica. Therefore, we needed to infer the topics for each review along with their probabilities. We then combined the sentiment lexica of different topics using the topic probabilities of the document as weights. For each n-gram in the document, we assigned a score based on the sentiment values of the generated lexicon (see Equation 4.6), skipping document terms not contained in this lexicon. We only computed sentiment values for terms contained in the training corpus, since for unseen terms no accurate prediction can be made. Finally, we summed up the sentiment values for all terms w in the review, normalized by the number of terms, and obtained a score for the whole document d:

$$\mathrm{CDSV}(d) \quad = \quad \frac{1}{|D|} \sum_{i=1}^{|D|} \mathrm{CDSV}(w, d) \tag{4.9}$$

Based on the CDSV values for each document, we can order the documents in a descending order from positive to negative and compare with the original ranking based on user ratings.

Table 4.3: Correlation of the automatic generated ranking with the rankings produced by the different users and inter-rater agreement measured through Kendall's W

User	**Spearman's ρ**
User 1	0.31
User 2	0.29
User 3	0.42
User 4	0.25
User 5	0.24

Avg. Spearman's ρ
0.30

Kendall's W
0.56

Quality Measures. The different methods provide a score for each review. By ordering the documents in the test set based on this score, we obtain a ranking from most negative to most positive rating for each method. This ranking is compared with the original ranking obtained through ordering the documents according to their star ratings. To account for the many ties in the original ranking (due to the 5 discrete star ratings), we use Kendall's τ_b to compare the computed rankings with the original one:

$$\tau_b = \frac{C - D}{\sqrt{C + D + T_x}\sqrt{C + D + T_y}} \tag{4.10}$$

where the different numbers of pairs are summed up and normalized with C number of concordant pairs, D number of discordant pairs, T_x number of pairs tied in the first but not in the second ranking , and T_y number of pairs tied in the second but not in the first ranking. The values of τ_b are in the range of -1.0 for perfect negative correlation, 0.0 for no correlation, and 1.0 for perfect correlation.

Additionally, we report accuracy values and show ROC curves and ROC-AUC values considering the review prediction problem as a binary classification task. All reviews with four or five stars are considered as being positive, whereas all reviews with one or two stars are regarded negative. Three-stars reviews were ignored. In the same setting, we show precision-recall curves as well as the precision-recall break-even points (BEPs) for these curves (i.e. precision/recall at the point where precision equals recall, which is also equal to the F1 measure, the harmonic mean of precision and recall in that case).

4.2.3 Results

We present results for the manual evaluation and the automatic evaluation based on review classification as described above.

User Evaluation. Table 4.3 shows the Kendall's W inter-rater agreement for the test users together with Spearman's ρ for each user. The average Spearman rank correlation coefficient is 0.30 and the average inter-rater agreement measured by Kendall's W is 0.56. The latter value indicates that the task was very hard for human judges, and there were many controversial cases where users did not agree on the ordering. Nevertheless, our algorithm achieves a correlation far better than random. These results are very promising, and we can build on them in future work. They were also statistically significant (t-test with 75 comparisons per user, significance level 0.01), indicating a clear relation between our automatic method and user judgments.

Review Classification. The overall results for our context-dependent sentiment lexicon (CDSL) and the comparison to baseline approaches are shown in Tables 4.4 and 4.5. Since we need a training corpus for the domain-dependent approaches, we evaluated

Table 4.4: Correlation of the original ranking with the rankings produced using CDSL and the baseline approaches

Approach	Kendall's τ_b		
	Blitzer	5-fold Cross-Validation	
	Test Set	Small Set	Large Set
MPQA Lexicon	0.326	0.324	0.313
SentiWordNet	0.345	0.312	0.327
PMI Small	0.328	0.293	n.a.
PMI Large	0.364	n.a.	0.341
CDSL Small	0.410	0.466	n.a.
CDSL Large	0.481	n.a.	0.513

Table 4.5: Accuracy comparing CDSL with other approaches

Approach	Accuracy		
	Blitzer	5-fold Cross-Validation	
	Test Set	Small Set	Large Set
MPQA Lexicon	0.669	0.650	0.653
SentiWordNet	0.687	0.679	0.681
PMI Small	0.683	0.667	n.a.
PMI Large	0.702	n.a.	0.687
Li et al. [118]	0.690	n.a.	n.a.
Denecke [53]	0.707	n.a.	n.a.
CDSL Small	0.732	0.685	n.a.
CDSL Large	0.775	n.a.	0.806

the results obtained on a small and on a large corpus (see Section 4.2.2) to see the influence of the training size on the results. For cross-validation we report the results on the respective training corpus. Table 4.4 reveals that the domain-dependent baseline (PMI) performs similar to the domain-independent lexica, MPQA and SentiWordNet, showing improvements when a larger training dataset is available. Our approach (CDSL) significantly outperforms the others – especially if the lexica are generated using the large training set. The performance of CDSL is even better on our Epinions dataset. The same holds for classification accuracy (Table 4.5). We consider our approach not fully supervised since we do not need new annotated training data for new datasets or domains. Compared with other entirely supervised approaches like Denecke [53], or unsupervised classification using existing lexica (e.g. Li et al. [118]), we improve the accuracy by approximately 10 percent points.

Figures 4.2 and 4.3 show the precision-recall curve for the binary classification task on the Blitzer test set where a review is considered positive if the rating is four or five, and negative for a rating of one or two. The break-even point (BEP) is with 0.775 for CDSL

Table 4.6: Break-Even Points and ROC-AUC values for CDSL and the baseline approaches

Approach	Blitzer Test Set		5-fold Cross-Val. Large Set	
	BEP	AUC	BEP	AUC
MPQA Lexicon	0.692	0.746	0.675	0.750
SentiWordNet	0.688	0.753	0.683	0.737
PMI Large	0.697	0.768	0.688	0.757
CDSL Large	0.775	0.849	0.808	0.882

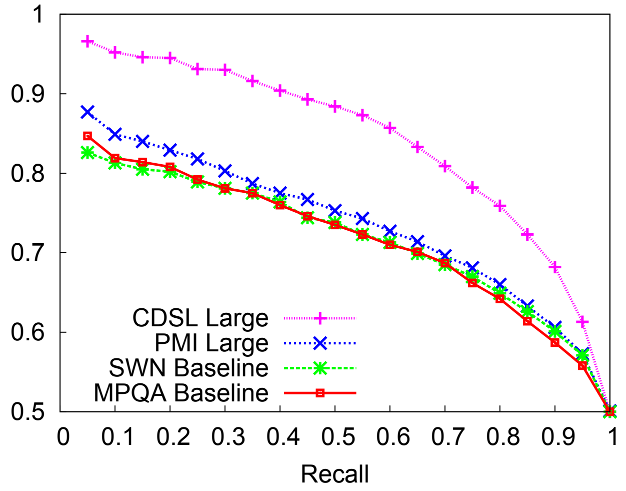


Figure 4.2: Precision-Recall Curves for the Blitzer Test Set

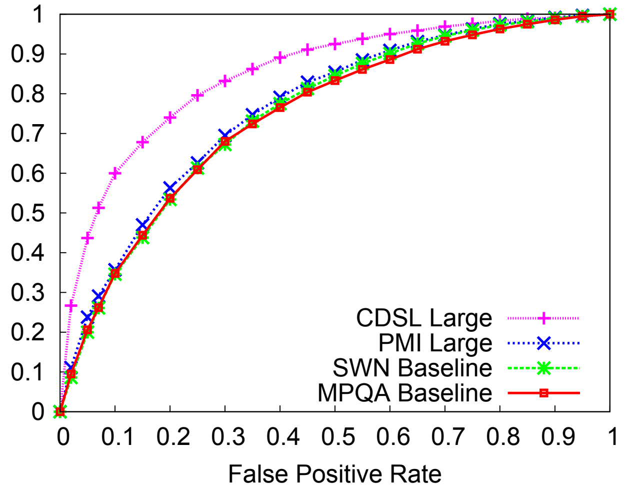


Figure 4.3: ROC Curves for the Blitzer Test Set

substantially higher than for the other approaches. The detailed BEP and ROC-AUC values are shown in Table 4.6. The results for the large dataset using cross-validation are shown in Figures 4.4 and 4.5. Our CDSL approach achieves a BEP of 0.808.

To evaluate the influence of the number of latent topics, we varied this number between 25 and 500 (cf. Table 4.7). We also experimented with using only one topic (i.e. basically making our lexicon a context-independent one). The table also shows results using the original Epinions categories instead of applying LDA. For each category, we built a lexicon and mapped the Amazon categories of the test set to the Epinions categories. We observe that our automatic topic extraction approach works better and does not rely on category information, which might not be available for other datasets like blogs or comments.

Table 4.8 shows the performance of our method for the different product categories in the test set. Performance varies with distinct categories, which might be due to their specific characteristics: Reviews about "Books" and "DVDs" also contain words about their content or plot which makes it difficult to find the specific latent topics representing books or DVDs. "Electronics" on the other hand subsumes many different products but

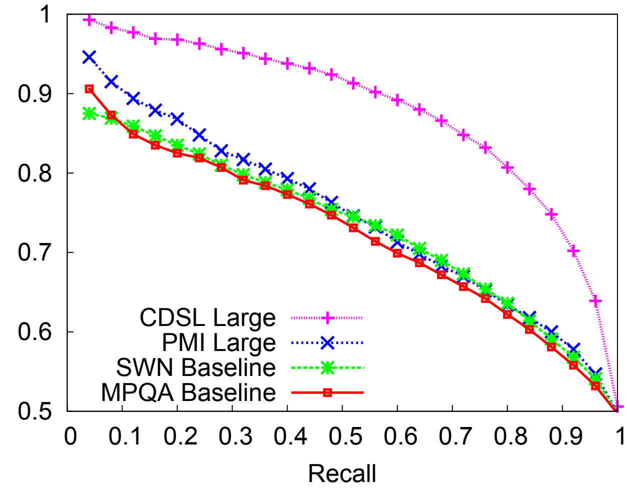


Figure 4.4: Precision-Recall Curves for 5-fold Cross-Validation on Large Dataset

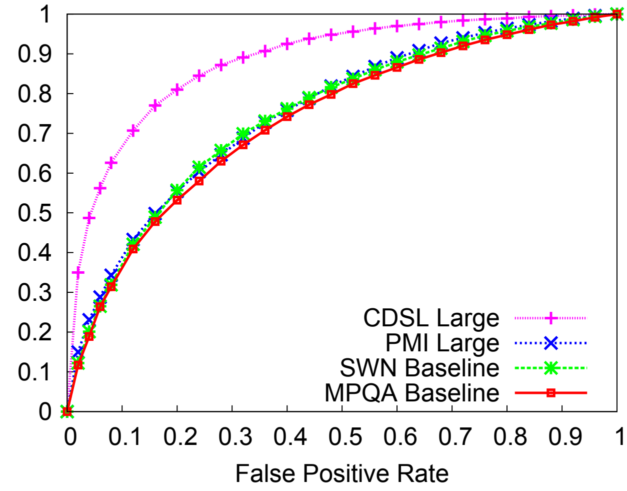


Figure 4.5: ROC Curves for 5-fold Cross-Validation on Large Dataset

Table 4.7: Accuracy and correlation of the original ranking with the rankings produced using our context-dependent sentiment lexicon (CDSL) using different numbers of latent topics for the Blitzer test set

Num. of Topics	**Kendall's τ_b**	**Accuracy**
Orig. Categories	0.203	0.611
1 (Context Indep.)	0.454	0.758
25	0.442	0.746
50	0.476	0.769
75	0.481	0.775
100	0.472	0.765
200	0.469	0.766
500	0.447	0.750

Table 4.8: Accuracy and correlation of the original ranking with the rankings produced using our context-dependent sentiment lexicon (CDSL) for the different categories of the Blitzer test set

Category	**Kendall's τ_b**	**Accuracy**
Books	0.417	0.755
DVDs	0.451	0.763
Electronics	0.527	0.792
Kitchen & Housewares	0.530	0.812

Table 4.9: Sample topic "fast food" with automatically assigned sentiment values, probability of each n-gram belonging to this topic and the average SentiWordNet scores

N-Gram	**POS**	**Topic**	**Sentiment**	**Average SWN Score**		
		Probability	**Value**	Positive	Negative	Objective
food	noun	0.035	-0.004	0.00	0.04	0.96
burger	noun	0.015	0.007	0.00	0.00	1.00
eat	verb	0.014	-0.020	0.04	0.00	0.96
sandwich	noun	0.009	0.029	0.00	0.00	1.00
meal	noun	0.007	0.013	0.00	0.00	1.00
restaurant	noun	0.006	-0.001	0.00	0.00	1.00
service	noun	0.006	-0.001	0.00	0.00	1.00
fast food	adjective,noun	0.006	0.001	0.08	0.06	0.86
cheese	noun	0.004	0.031	0.00	0.00	1.00
mcdonalds	noun	0.004	-0.063	–	–	–
burger king	noun,noun	0.003	-0.005	0.01	0.00	0.99
fresh	adjective	0.002	0.056	0.16	0.27	0.57
french fry	adjective,noun	0.002	0.037	0.00	0.00	1.00
tasty	adjective	0.001	0.054	0.62	0.25	0.12
cold	adjective	0.001	-0.056	0.15	0.37	0.48
kfc	noun	0.001	-0.110	–	—	–
variety	noun	0.001	0.073	0.15	0.08	0.77
grease	noun	0.001	-0.103	0.00	0.06	0.94
atmosphere	noun	0.001	0.066	0.00	0.00	1.00
fast	adverb	0.001	0.047	0.00	0.00	1.00
hot dog	adjective,noun	0.001	0.074	0.09	0.14	0.77
dirty	adjective	0.001	-0.139	0.08	0.47	0.45

the words in the reviews make it easier to put them in the right context.

Finally, an example entry from one of our generated topic lexica is shown in Table 4.9. These are the "pure" topic sentiment scores without combining them with a general, context-independent lexicon. Apart from the sentiment value assigned by our algorithm also the SentiWordNet scores are listed. Note that our approach also assigns sentiment values to named entities and terms that do not appear in a common English dictionary. For example "McDonald's" obtains a negative sentiment value in the context of the "fast food" topic whereas "Burger King" seems to be perceived more neutral. This could be due to the reputation McDonalds has in the U.S. for selling low quality, unhealthy food. To compare the scores for n-grams, we also tried to compute sentivalues for them by averaging over the sentivalues of each term (this explains the score for "Burger King"). The adjective "dirty" and the noun "grease" in have high negative values, whereas "atmosphere" and "fresh" get high positive values. Also note that the adjective "fresh" on average is considered rather negative in the context-independent SentiWordNet, whereas we assign a positive value in the fast food context.

4.3 Related Work

Sentiment analysis and opinion mining are highly active research fields. Pang and Lee [143] give a good introduction and overview of the field. In the following we will look at some aspects in detail.

Sentiment Lexica. There exist various domain-independent sentiment lexica, one of the most prominent being SentiWordNet [56], which was recently extended by exploiting the graph structure of the underlying WordNet lexicon using Page-Rank-like propagation of sentiment values [8]. Manual sentiment annotations can be found in the MPQA corpus [194]. Whitelaw et al. [193] created a lexicon consisting of around 2000 manually selected, general sentiment terms. Turney and Litman [183] make use of a small set of seed terms with positive and negative semantic orientation, and estimate the polarity of new terms by computing co-occurrence based statistics (using pointwise mutual information and latent semantic indexing). Some of the technical components in our approach resemble the ones used by Turney and Litman but, in contrast, are applied in our work to identify topical context and exploit numeric ratings in reviews. In our experiments, we will show comparisons to the above mentioned lexica.

Context-Dependent Sentiment Lexica. There is also work on generating sentiment lexica that take context and topic information into account. The problem of polarity shift of adjectives in certain domains [137] inspired Fahrni and Klenner [57] to identify domain-specific nouns, and create specific sentiment lexica of adjectives for these target nouns. Wikipedia is used to find the nouns and polarity is estimated through a bootstrapping approach for extracting patterns. Kanayama and Nasukawa [97] identify opinion terms using a seed set of general polarity terms and their connections to other entities in reviews in order to discover new polarity terms. They achieve domain orientation by applying their methods separately on discussion board corpora from different domains. Similarly, starting with a seed of sentiment words, Qiu et al. [150] iteratively expand sentiment lexica through connection to other terms for separate review corpora on categories like “digital cameras”, “DVD players” or “cell phones”. Also starting with a seed set of sentiment terms, Jijkoun et al. [95] extract syntactic patterns and potential targets from a background corpus. They compare the frequency of occurrences in the background corpus with the frequencies in a topic-specific set of documents using chi-square. The topic-specific corpus is obtained by querying a corpus with a topic keyword. For the top targets, sentiment terms are then extracted for the topic-specific lexicon. In their work [28], Bross and Ehrig analyze review data with a specific *pros* and *cons* structure to identify the polarity of opinion tuples (o, p) of opinion words o and entities p (e.g. (“intuitive”, “menu”)) by exploiting the correlation of their occurrence in the *pros* and *cons* lists. In contrast to the described works, the topical context studied in our approach is placed in between a small number of large and fixed domains and very fine-grained entity specific sentiment assignments. Furthermore, our techniques are orthogonal to the described ones in the sense that we make use of numeric ratings accompanying review articles. The work by Li et al. [118] studies a joint model of latent topics and sentiment using a seed set of manually selected terms; however, they do not exploit information from review ratings to build their lexica. We compared to this approach in our experiments, and showed that considering such ratings leads to a substantial performance boost.

Context-Dependent Sentiment Analysis. There are several works that analyze sentiment in the context of topics and aspects. Choi et al. [37] analyze corpora based on queries related to different domains like "business" or "politics" separately for each of these domains to improve sentiment oriented search. Lu et al. [125] describe a method for summarizing the opinion on product aspects from a set of short comments. They employ a three-phase approach in which they first extract latent topics using PLSI, then classify the sentiment of short comments, and finally aggregate over the obtained sentiment scores. Titov and McDonald [179] focus on the first of these steps, and extract multi-grain topics from online reviews. They apply an LDA-like approach coined MG-LDA to distinguish between global topics (e.g. "London") and ratable aspects within these topics (e.g. "transportation"). In contrast, we focus on the task of automatic sentiment lexicon generation.

Sentiment Classification. Finally, there is a plethora of work on sentiment classification and rating prediction. Sentiment classification (described, for instance, in [144, 176]) deals with the problem of automatically assigning opinion values (e.g. "positive" vs. "negative" vs. "neutral") to documents or topics using various text-oriented and linguistic features. Recent work in this area makes also use of SentiWordNet to improve classification performance [53]. Qu et. al [151] apply regression models learned on training sets of review-rating pairs to predict product ratings. Blei and McAuliffe [24] modify latent topic models to consider information from additional response variables (e.g. rating scores of product reviews), and use their models for rating prediction. Cross-domain sentiment classification was studied, for instance, in [142] where spectral graph analysis is used to infer links between domain-independent and domain-specific terms. Similar to many of the aforementioned works we borrow from latent topic modelling, especially latent Dirichlet allocation (LDA) [25], as well as discriminative feature analysis and selection [134, 204]. However, the focus of our work is on the generation of topic-specific sentiment lexica rather than on predictions or summaries of ratings and sentiments.

Our Contributions. We presented an approach for automatically assigning sentiment values to terms based on the topical context of a term. We made use of latent topic analysis to identify topics in a corpus and generate sentiment lexica for these topics. In what is a novel approach we exploited user-generated reviews and star ratings for products to build these lexica. Our experiments on a large dataset of product reviews show that topical context is important for assigning sentiment values to terms and that our approach to lexicon generation can capture topic dependency. Classification experiments revealed that our method outperforms other state-of-the-art sentiment lexica. An additional manual user evaluation showed promising results, indicating a clear relation between the automatically generated sentiment values and judgments by human assessors.

CHAPTER

FIVE

DIVERSIFICATION OF RANKINGS

In this chapter we focus on the diversification of Web search result rankings and product review rankings. We show that for both tasks latent Dirichlet allocation can be employed to improve topical diversity within the top-k entries of a ranking.

5.1 Introduction

Rankings are a common way to present results to users in the Web. Especially when many interesting results are available, rankings provide a means for guiding the user in selecting information. These rankings are generated by ordering a result set taking various objectives into account. Web search result rankings are a well studied phenomenon having gained popularity through Web search engines from Altavista, Google, Yahoo!, or Microsoft. Product review rankings on the other side have not received much attention so far. For both types, diversification is an important aspect. While for Web search engines relevance to a query is the main factor when computing a result ranking, this does not apply for product review rankings.

Web Search Results. Information Retrieval aims to provide the best possible results to meet a user's information need. Although keyword search and relevance rankings have been proven to be powerful tools to identify a user's interest and produce result lists with relevant pages, these mechanisms fail in certain situations. Ambiguous query terms are examples where relevant documents can not be reliably assessed without additional information from the user. Systems have to estimate a suitable relevance score and then rank accordingly. The most common way to produce these rankings is to follow the probability ranking principle (PRP) [160], which favors documents that are more likely to contain relevant information. For queries where the relevance scores for documents entail a lot of uncertainty, relevance rankings tend to leave a great deal of users unsatisfied: they abandon the query. Result diversification can reduce this effect [45, 155] significantly.

Ambiguous queries are not the only reason why search engine results should reflect diversity. Queries like "Napoleon" or "immigration" are less ambiguous but rather multifaceted. To capture different aspects of such queries a result set must contain diverse information and avoid semantically similar content within the top-k results. A truly diverse ranking then also offers an overview of the whole topic including various aspects and views.

Ideally, Web search results are not biased towards a certain interpretation or aspect. However, depending on the algorithm for assigning relevance scores certain interpretations of queries may be represented disproportionally high within the result set. For ambiguous queries such as "jaguar" or "java", commercial search engines nearly exclusively present documents about one interpretation ("car" and "programming"). Reducing the influence of the (manipulable) relevance score by combining it with a diversity aware component can help more users find what they are looking for.

As described by Wang and Zhu [188], diverse rankings can be seen as the result of ranking under uncertainty where the user's information need cannot be ultimately defined. In the context of ambiguous queries, a system has to make a trade-off between the relevance of an isolated document and the risk involved of missing relevant aspects of a query. This task is tackled by Wang and Zhu by applying Modern Portfolio Theory [128], which is an economic theory that describes how to minimize the risk by not "putting all one's eggs in one basket", but on different investments. For ranking, this means to not favor one interpretation or aspect of a query over all others but prefer a *diverse* ranking.

Although diversifying Web search results has recently attracted a lot of interest within the research community [156, 188, 2, 69], automatic evaluation of diversity is still an open problem. Following [38], the TREC community has designed a task for subtopic retrieval in 2009 within the Web track [39]. The evaluation is based on subtopics of a query. These were identified using a query log of a commercial search engine and co-clicks, related queries and other information to find the different users' information need for each query. This also includes some manual judgement of the extracted subtopics. One of the drawbacks of this approach is the rather sparsely annotated data which makes it difficult to use for judging commercial search engines' results. The extraction process for the subtopics is also susceptible to missing aspects/subtopics of a query. The major drawback, however, is the need for manual judgement of whether a given Webpage covers a subtopic sufficiently or not. These judgements are cumbersome and costly.

In Section 5.2 we present a topic-centered approach for evaluation in contrast to the user-centered approach used in TREC. We propose an evaluation framework based on the Wikipedia encyclopedia and evaluate the diversity of Web search results for queries derived from titles of Wikipedia disambiguation pages. The coverage of the different aspects for a query present in Wikipedia is quantified using different entropy-based measures. We compare this evaluation setting with the TREC evaluation framework and show that we get comparable results with less costs and without having access to a large query log.

In addition, we present an approach to diversify search results by reranking. We estimate the relevance score of a document by its position in the original ranking and introduce a second score to reflect the additional diversity the document could add to the result list. This score is based on the variance of the underlying model for the document representation. We investigate language models and topic models [25], which have been shown to be useful document representations in the context of information retrieval tasks [209, 190].

We show that the variance-based reranking outperforms the original rankings of two large commercial search engines with respect to diversity within the top-k results. Moreover, we show that latent topic models achieve competitive diversification requiring significantly less reranking. By comparing the proposed Wikipedia-based evaluation framework with the TREC subtopic retrieval evaluation we see comparable results without the need for a large-scale manual annotation effort.

Consumer Product Reviews. It has become a routine among on-line and off-line consumers to inform themselves on review Web sites before purchasing a certain product. This has given rise to a considerable amount of customer reviews on e-commerce Web sites. To this end potential customers usually browse through a lot of on-line reviews in order to build confidence in a particular item prior to purchasing it. While reviews have become an important factor in helping Web crowds to further assess the quality of products on-line, increasing volume of reviews themselves has led to an information overload [35]. Popular products have thousands of reviews. While excess of reviews is a growing problem, recommending unbiased and helpful reviews is a growing research field. The quality of reviews may vary drastically [139] and might mislead potential buyers. Such humongous amount of information not only distracts the confidence seeker, it might hinder the original goal of users in the first place: They will give up buying a certain product. To deal with these problems, review recommendation techniques are proposed. Review recommendation involves implementing machine-learning techniques for analyzing the product reviews based on their lexico-semantic features in order to classify the reviews and recommend balanced and useful reviews to the readers.

While review recommender systems aim at automatically classifying reviews, some commercial Web sites such as Amazon[1] and TripAdvisor[2] approach this problem by allowing users to rate the reviews using star ratings to improve the rankings (e.g. *this review was helpful vs. not helpful*). There are two inherent problems to these ranking based on user feedback: First, good objective reviews contain quite likely redundant information and ranking them based on the helpfulness score will not cover all aspects. Second, these Web sites do not take into account the personal bias. Not all reviews are helpful to everybody. Due to the fact that different users put different emphasis on different aspects, (e.g. *I don't care about battery life, but really need lots of memory*), helpfulness can only be used to filter out very bad reviews. Therefore researchers are increasingly distinguishing between the task of review recommendation [1] and review ranking [63]. To improve existing review recommendation techniques and at the same time improve the ranking used for evaluating helpfulness merits of existing reviews, we propose a novel approach to model and rank reviews. The two main components of our system rely on latent Dirichlet allocation (LDA) to model the reviews and on Kullback-Leibler divergence to generate an adequate ranking. We make use of the assigned star rating for the product as an indicator of the polarity expressed in the review towards the latent topics. Our framework covers different ranking strategies based on users' needs to adapt to various user scenarios. We currently support three strategies to summarize all reviews, to focus on a particular latent topic, or to focus on positive, negative or neutral aspects. We evaluated the system using manually annotated review data gathered from a popular review Web site.

[1] `http://www.amazon.com`

[2] `http://www.tripadvisor.com`

5.2 Language Models and Topic Models for Web Search Result Diversification

In this section we present our approach to diversify Web search results by reranking. We compare the use of language models and topic models as underlying representation for the task and introduce Kullback-Leibler divergence as evaluation criterion. Further we propose and evaluate a diversification framework based on Wikipedia.

5.2.1 Approach

Our goal is on the one hand to cover for each query as many *different aspects* as possible within the top-k search results. On the other hand, ranking of Web pages is predominantly done by picking the most *topically relevant*[3] pages for a keyword query according to the probability ranking principle [160]. A diverse search result cannot neglect the relevance aspect. Thus, the relevance of Web pages for a user's query still plays an important role. A trade-off between relevance and diversity [40] is incorporated within our system to accommodate this mutual relation.

In its most simple form the probabilistic ranking principle assumes that the relevance of each individual result only depends on the query, but does not depend on the other results. Under this assumption, given a good estimate of the relevance $E(r_i)$ for each result r_i, ordering the results by decreasing $E(r_i)$ is optimal. However, especially for Web search results, this assumption clearly does not hold. In the extreme, if the most relevant result is duplicated, the top results will all be the same. More generally, if results overlap with each other, the top results will often be pre-occupied by one interpretation of a query. Thus, the general goal of diversification is to balance between relevance of individual results and their overlap.

One popular approach to this end is to minimize the mutual overlap between the top k results, using some similarity measure such as Jaquard similarity or cosine similarity. We adopt a closely related approach, originally introduced in [188], which *maximizes the expected relevance* $E(R_k)$ and *minimizes the variance* $Var(R_k)$ for the top k documents of a search result $R_n = r_1, ..., r_n$:

$$E(R_k) - B * Var(R_k) \tag{5.1}$$

where B regulates the trade-off between relevance and diversity. Expected relevance $E(R_k)$ and variance $Var(R_k)$ are calculated as weighted sum over the individual results r_i:

$$\begin{array}{ll} E(R_k) & = \sum_{i=1}^{k} w_i E(r_i) \\ Var(R_k) & = \sum_{i=1}^{k} \sum_{j=1}^{k} w_i w_j c_{i,j} \end{array}$$

where $c_{i,j}$ is the covariance of results r_i, r_j (see Equation 5.7), and w_i is a normalized discount factor [92]:

$$w_i = \frac{1}{log_2(i+1) \sum_{j=1}^{n} \frac{1}{log_2(j+1)}} \tag{5.2}$$

$\frac{1}{log_2(i+1)}$ is 1 for rank $i = 1$ and decreases monotonically, the second factor in the denominator normalizes the sum of all w_i to 1.

[3] Possibly, commercial search engines also include popularity and other factors in their ranking.

Diversity is inversely proportional to variance: A small variance $Var(R_k)$ corresponds to large diversity, because all diverse aspects of a query are covered more or less equally, e.g., when the aspects follow a uniform distribution, the variance is 0. B controls the relative importance of diversity vs. relevance. For $B > 0$ relevance and diversity are balanced against each other. In particular for ambiguous queries, choosing relevant and at the same time diverse and complementary documents with high $E(R_k)$ and low $Var(R_k)$ minimizes the risk that the top k results do not contain any relevant document at all for some of the possible query interpretations. For $B < 0$ relevance and variance are maximized, and thus diversity is minimized. This favors one particular interpretation with high $E(R_k)$ but also high $Var(R_k)$, which increases the risk of missing out other plausible interpretations altogether.

Finding a reranking that globally optimizes the objective in Equation 5.1 is infeasible, as it would require testing all permutations of the original ranking. Thus, following common practice, we approximate the optimal reranking using a greedy algorithm that selects for each new rank k the result r_i such that the increase in the objective at rank k $(O_k - O_{k-1})$ is maximized [188]:

$$\begin{aligned} O_k - O_{k-1} = \quad & \textstyle\sum_{i=1}^{k} w_i E(r_i) - B \sum_{i=1}^{k} \sum_{j=1}^{k} w_i w_j c_{i,j} \\ & \textstyle - \sum_{i=1}^{k-1} w_i E(r_i) + B \sum_{i=1}^{k-1} \sum_{j=1}^{k-1} w_i w_j c_{i,j} \\ = \quad & \textstyle w_k(E(r_k) - B w_k c_{k,k} - 2B \sum_{i=1}^{k-1} w_i c_{i,k}) \\ \propto \quad & \textstyle E(r_k) - B w_k c_{k,k} - 2B \sum_{i=1}^{k-1} w_i c_{i,k} \end{aligned} \tag{5.3}$$

The multiplier w_k is constant for all candidate documents to be selected for rank k and thus can be ignored.

In contrast to [188] we do not estimate the expected relevance $E(r_k)$ from the query and individual results, but rather rely on the original ranking of the search engine, which takes into account a variety of factors, including relevance, popularity, and user preferences. As search engines typically do not provide an actual score we set $E(r_k)$ to the discount factor w_i of a result document r_i to be reranked to position k. $c_{k,k}$ is the (inner) variance $\sigma^2(r_k)$ of result r_k at the new rank k. This leads to the following optimization objective: At each new rank k select the document r_i at the original rank i, such that

$$w_i - B w_k \sigma^2(r_i) - 2B \sum_{j=1}^{k-1} w_j c_{j,i} \tag{5.4}$$

is maximized.

A couple of technical statements are in order: To effectively balance $E(R_k)$ and $B * Var(R_k)$ they should be in the same order of magnitude. To this end, we calibrate B as follows:

$$B = \frac{\beta}{avg_i \sigma^2(r_i)} \tag{5.5}$$

where $avg_i \sigma^2(r_i)$ is the average (inner) variance over all results r_i. With this approach, $\beta = 1$ gives approximately equal weight to relevance and diversity[4].

The complexity of the greedy reranking algorithm is $O((n-k) * k * |V|)$ for reranking in the top-k results, given n overall results and vocabulary size $|V|$. Thus for relatively

[4] With $\beta = 1$, $\sum_{i=1}^{n} O_i \approx \sum_{i=1}^{n} w_i - \frac{\beta \sum_{i=1}^{n} w_i \sigma^2(r_i)}{avg_i \sigma^2(r_i)} = 0$. This assumes that the overall sum of covariances is zero, which is probably an underestimation.

small k in the range of the typical 10 results on the first page online reranking is feasible, in particular, when combined with standard techniques such as caching popular queries.

Representation of Documents. In order to calculate the variances we represent individual documents r_i as vectors. We have experimented with two alternative representations: Smoothed (unigram) language models and latent topic models.

Language Models. The Jelinek-Mercer smoothed language models [209] for a document r are defined as

$$q_i = \lambda * p(v_i|r) + (1 - \lambda) * p(v_i) \tag{5.6}$$

where $p(v_i|r)$ is the relative frequency of term v_i in r, and $p(v_i)$ is the relative collection frequency of v_i. For smoothing we use the relatively large[5] $\lambda = 0.99$.

Given two vectors $U = u_1 \dots u_n$ and $Q = q_1 \dots q_n$, their co-variance is defined as:

$$\begin{aligned} Var(U, Q) &= \frac{1}{n} \sum_{i=1}^{n} (u_i - \bar{u})(q_i - \bar{q}) \\ &= \frac{1}{n} \sum_{i=1}^{n} u_i q_i - \frac{1}{n^2} \end{aligned} \tag{5.7}$$

The simplification is based on $\bar{u} = \bar{q} = 1/n$.

It is interesting to compare this to cosine similarity used by other approaches to diversification:

$$Cos(U, Q) = \frac{\sum_{i=1}^{n} u_i q_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \sqrt{\sum_{i=1}^{n} q_i^2}}$$

As can be seen, covariance and cosine similarity differ only w.r.t. normalization, which plays a minor role when operating on vectors representing a normalized probability distribution. However, whereas minimizing the mutual cosine similarity between results only accounts for the overlap between results, minimizing the overall variance of a result list also accounts for the inner variance of individual results. Thereby, results that cover more aspects of a query will tend to be ranked higher.

The probability of a word v in a document r is defined as:

$$p(v|r) = \frac{n(v, r) + \mu * p(v|R_n)}{|r| + \mu} \tag{5.8}$$

where $n(v, r)$ is the count of word v in r, r is the document length, $p(v|R_n)$ is the relative frequency of word v in the entire collection R_n (aka collection frequency), and μ is a hyperparameter controlling the degree of smoothing.

For calculating $E(r_i)$ we have two alternatives: One is to simply take into account the original ranking given by the search engine, e.g., by setting $E(r_i) = w_i$. The other alternative is to use the maximum likelihood estimate: $E(r_i) = \prod_{i=1}^{|V|} p(v_i|r)^{q_i}$, where q_i is the vector representation of the query (in our case typically 1 or 0 at least for the baseline). These two estimates may also be combined by linear interpolation.

[5] Since we need smoothing only for avoiding zero probabilities in our evaluation based on Kullback-Leibler divergence, we have chosen an unusually large λ. Smaller λs would just make the individual documents more similar, and thereby reduce their (co-)variance.

Table 5.1: Top 10 search results for query “Caesar” using Google search engine F. Note that topic Julius Caesar also covers a variety of sub-topics.

Rank	Original
1	Caesars Palace Hotel
2	Caesars Palace Hotel Shopping
3	Caesars Windsor Hotel
4	Gaius Julius Caesar Biography
5	COADE CAESAR II - Pipe Stress Analysis
6	COADE Company
7	Free Website on Community Architect
8	Julius Caesar: Guide to Online Resources
9	Caesar Augustus: Guide to Online Resources
10	Caesar Miniaturs Company

Table 5.2: Top 10 diversified search results for query “Caesar” using LDA and LM based on Google results

Rank	LDA	LM
1	Julius Caesar Biography	A Weblog by Julius Caesar
2	Caesar III Heaven Games	Julius Caesar Biography
3	Free Website on Community Architect	Caesar’s Campaigns in Gaul
4	Shakespear’s Julius Caesar	Commentariorvm de bello gallico
5	A Weblog by Julius Caesar	Shaw’s Caesar and Cleopatra
6	COADE CAESAR II - Pipe Stress Analysis	Augustus Biography
7	Littel Caesar Movie 1930	CAESAR Anthropometry
8	Commentariorvm de bello gallico	Littel Caesar Movie 1930
9	Caesar’s Campaigns in Gaul	Julius Caesar Biography
10	Shakespear’s Julius Caesar Paraphrase	Svetoni tranqvilii vita divi ivli

Latent Dirichlet Allocation. Smoothed language models may suffer from the curse of dimensionality, and thus not properly represent the topics or aspects of a result list. As a consequence, variance measured directly on the bag of words may not be a good indicator for topical coverage. For example, if two results are about the same topic, but use different vocabulary, their covariance will be underestimated.

Thus as an alternative representation, we have also experimented with latent Dirichlet allocation, which maps documents to a mixture of only a few latent topics. Variance is then estimated on the much lower dimensional representation of the latent topics $P(z_i = j \mid d_i)$ as defined in Equation 2.8 rather than on the bag of words derived from Equation 5.6. A whole document can thus be represented as a mixture of different topics. For Web pages where the author of a document can be considered one entity, these topics reflect the entity’s view of this document and her particular vocabulary.

5.2.2 Evaluation

To evaluate our approach we propose to use Wikipedia as a source of ground truth for diversity. Wikipedia has been shown to be an effective and reliable source of semantic knowledge [60] and was used before in the context of diversity evaluation [69]. We think that this kind of evaluation is superior to manually selected corpora to judge the diversity of Web search result rankings. Hand-crafted collections like the one used for the TREC subtopic retrieval task are not as complete and representative as a community maintained encyclopedia like Wikipedia.

Table 5.3: Wikipedia pages containing the query "Billboard" and its corresponding link indegrees.

Titles of Wikipedia pages containing "billboard"	**Link Indegree**
Billboard magazine	2100
Billboard Hot 100	932
Billboard 200	323
Billboard (advertising)	74
Billboard Music Award	24
Adult Contemporary (Billboard Chart)	16
Billboarding	2
Billboard Liberation Front	2
Billboard antenna	2
Billboard toppling	1
List of most frequently mentioned brands in the Billboard Top 20	1
Billboard Utilising Graffitists Against Unhealthy Promotions	0
Billboard Comprehensive Albums	0
Billboards of Lahore	0

For the evaluation, we compare the original ranking with the diversity-oriented reranking. The test queries are taken from the titles of Wikipedia disambiguation pages. The basic assumption is that Wikipedia articles cover the major alternative interpretations of ambiguous queries. This claim was recently backed by [164], who showed that more than 50% of pages in their test set can be assigned to Wikipedia pages representing a particular sense of the query. Moreover, we also compare the various rankings with the "complete" search result returned for each query.

We conducted several experiments to evaluate our reranking algorithm and to verify our evaluation approach:

1. Reranking based on language models of search results.

2. Reranking based on topic models derived from latent Dirichlet allocation.

3. Comparing diversity of result rankings from Google and Yahoo!.

4. Comparing our evaluation using TREC data and manual judgement.

Data. To evaluate diversity we are interested in queries that have a broad variety of aspects. This does not necessarily mean that the queries are ambiguous. A keyword query like "Las Vegas" might have different meanings but even the interpretation as the name of a city has a lot of aspects and subtopics which diversity aware search engines should cover in the top-k results.

The generation of the ground truth test data was a two phase process. Firstly, we took the Wikipedia disambiguation pages and removed all pages containing digits in the title (e.g. Wikipedia page "442_(disambiguation)"). Secondly, we searched in a Wikipedia MYSQL-dump with the title of the disambiguation page in the title field of the database. All titles returning between 10 and 100 Wikipedia pages were kept and the others discarded. We sorted the titles of the disambiguation pages by the sum of the inlink degree of the Wikipedia pages. The top 240 titles constitute our query set and the corresponding Wikipedia pages are our ground truth data. One example query ("billboard") with its corresponding Wikipedia page titles is shown in Table 5.3.

To get the ranking of the commercial search engines from Google and Yahoo! we crawled[6] the result lists for each query up to rank 1000. For the Yahoo! search engine we got an average of 628 results per query; for Google we got 730. Search results from Wikipedia were discarded, in order to assess original and diversified rankings of non-Wikipedia results. Also all pages without textual content were removed from the collection. In addition, we removed boilerplate text from the result Web pages using boilerpipe[7], an open source library for extracting fulltext from HTML pages, to obtain clean content for each page. For both search engines we got the relevance rankings for each query ordered by rank. For Wikipedia as well as for the search engine results we removed stopwords.

For each rank k we define R_k as the concatenation of all documents $r_i, 1 \leq i \leq k$. The smoothed language model Q for each R_k is computed as described in Equation 5.6.

The language models U for Wikipedia are in addition weighted by the logarithm of the indegree of articles (d_j), in order to push more prominent interpretations[8]

$$p(v_i|W) = \frac{\sum_{j=1}^{m} log_2(d_j) * n(v_i, w_j)}{\sum_{j=1}^{m} log_2(d_j) * |w_j|} \tag{5.9}$$

where m is the number of Wikipedia result pages for a query, $n(v_i, w_j)$ is the frequency of word v_i in article w_j, and W signifies that the language model is conditioned on Wikipedia.

Unless otherwise noted, Q refers to the language model of top-k search results, U refers to Wikipedia articles, and S refers to the complete search result of a particular query.

Quality Measures. As a measure for how well the top-k Web search results for a query approximate the corresponding[9] Wikipedia articles we calculate the Kullback-Leibler divergence between the smoothed unigram language models for the top-k results and for Wikipedia articles. This measure estimates the number of additional bits needed to encode the distribution $U = u_1 \ldots u_n$, using an optimal code for $Q = q_1 \ldots q_n$, where $n = |V|$ is the combined vocabulary size.

$$\begin{aligned} D_{KL}(U||Q) &= H(U;Q) - H(U) \\ &= \textstyle\sum_{i=1}^{|V|} u_i * log_2(\frac{u_i}{q_i}) \end{aligned} \tag{5.10}$$

In our setting, distribution Q is the combined language model of the top-k search results and thus $D_{KL}(U||Q)$ can be directly used to measure the similarity with the combined Wikipedia articles and assess the coverage of the top-k Web pages with respect to Wikipedia.

To assess the effect of diversification on the search results Q, we also measure the entropy $H(Q)$ for the different rankings. The higher the entropy of the top-k results, the more diverse is the set of top-k Web pages.

$$H(Q) = -\sum_{i=1}^{|V|} q_i * log_2(q_i) \tag{5.11}$$

Spearman's rank correlation coefficient ρ is used to quantify the degree of reranking between two rankings x and y.

$$\rho(x, y) = 1 - \frac{6\sum_{i=1}^{n}(x_i - y_i)^2}{n(n^2 - 1)} \tag{5.12}$$

[6] For Google we used screen scraping; for Yahoo! we used the API; both were crawled in January 2010

[7] `http://code.google.com/p/boilerpipe/`

[8] This follows the observation in [164] that the indegree in Wikipedia correlates with the overall frequency of an interpretation.

[9] weighted combination of language models of Wikipedia pages returned by the Wikipedia search engine for a query

where x_i and y_i are the ranks at position i, and n is the number of results. A value of 1.0 means perfect correlation, 0.0 no correlation and -1.0 perfect negative correlation. In our setting, we are interested in the degree of reranking performed by the different algorithms with respect to the original ranking.

5.2.3 Results for Diversification Method

In this section we present the results for the diversification method. We compare language models and topic models with the original ranking and the optimal ranking.

An Example for Diversification. To exemplify the effect of diversification, we randomly picked the query "Caesar" from our evaluation set. Table 5.1 gives the top 10 results for this query by the original ranking and Table 5.2 by rankings diversified on the basis of latent topic models and language models[10]. The pages reflect a broad coverage of the query. While the original ranking covers some aspects of this query, including the historical persons "Julius Caesar" and "Caesar Augustus", hotels named "Caesar", and other companies using the iconic label "Caesar", both diversified rerankings arguably cover also other aspects, including movies and dramas about "Caesar", pointers to Julius Caesar's literary work, and also a broader variety of companies labeled "Caesar", with the notable exception of hotels. For other queries we can observe a similar effect. Generally, the diversified reranking achieves a better topic coverage in the top 10 results compared with the original ranking.

Diversification by LM vs. LDA. The goal of our first evaluation is to assess the effect of diversification for the two proposed models. Figures 5.1 and 5.2 show the Kullback-Leibler divergences $D_{KL}(U||Q)$ and $D_{KL}(Q||U)$ between the aggregated Wikipedia language model U and various rankings Q for ranks $k = 1..51$, averaged over all 240 queries in our test set. The original ranking from Google is labeled *orig*, *lda* denotes the topic model, and *lm* the language model based reranking. For the latent topic model *lda* we used 1000 topics. For the language model *lm* we used $\lambda = 0.99$ for smoothing. For the "optimal" ranking *opti*[11], we greedily reranked search results such that $D_{KL}(U||Q)$ is minimal for each rank k. For reranking we used $\beta = 1$, balancing relevance and diversity evenly.

As is to be expected, the original ranking *orig* has the largest divergence D_{KL} to Wikipedia in both directions, and the optimal ranking *opti* has the smallest divergence. The diversified reranking using language models *lm* slightly outperforms latent topic models *lda* at all ranks. However, this comes at the cost of a significantly larger amount of reranking: The average Spearman's rank correlation coefficient ρ between *lda* and *orig* is 0.23, which is more than twice of $\rho = 0.09$ between *lm* and *orig*[12] Interestingly, also the "optimal" reranking *opti* has a significantly higher $\rho = 0.17$.

Figure 5.3 shows how quickly the various rankings approximate the language model of the overall search result for each query. The smaller the divergence for the top k results the better they represent the overall result. Again, the original ranking *orig* shows the highest divergence. But the optimal ranking *opti* is surpassed by *lm* at rank 12, and by *lda* at about rank 25. Thus, optimizing w.r.t. Wikipedia content of a query generally also achieves a better representation of the search result in the first few ranks, but the generic diversification by minimizing variance performs slightly better for higher ranks (The plot for $D_{KL}(Q||S)$ is very similar).

[10] In all rankings the Wikipedia entries have been removed.

[11] "Optimal" refers to the optimal ranking with respect to Wikipedia coverage.

[12] The difference is significant with a confidence of 0.99 based on a paired t-test.

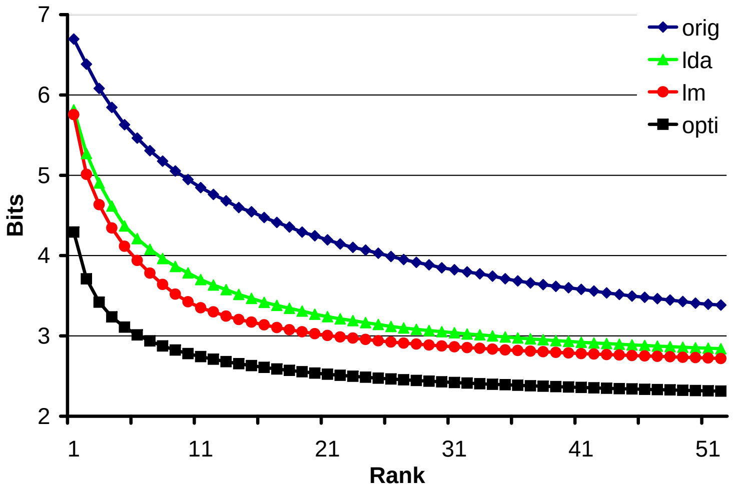


Figure 5.1: Kullback-Leibler divergence $D_{KL}(U||Q)$

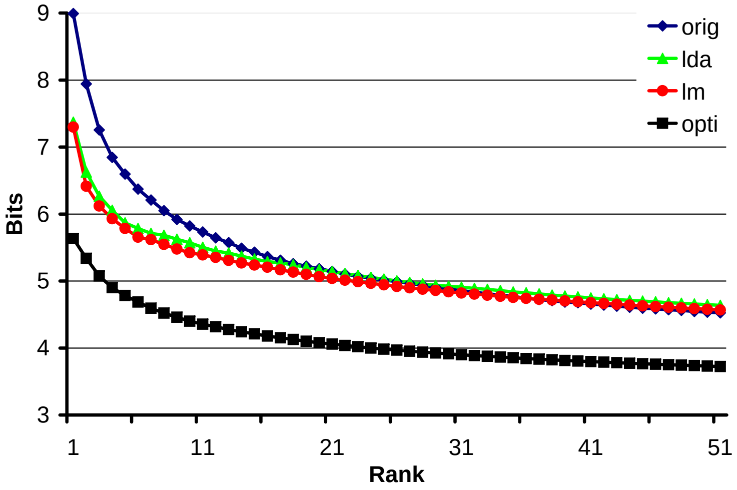


Figure 5.2: Kullback-Leibler divergence $D_{KL}(Q||U)$

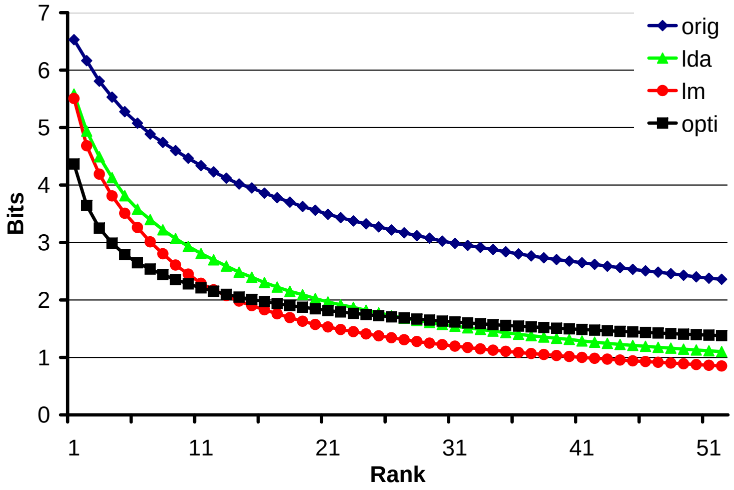


Figure 5.3: Kullback-Leibler divergence $D_{KL}(S||Q)$

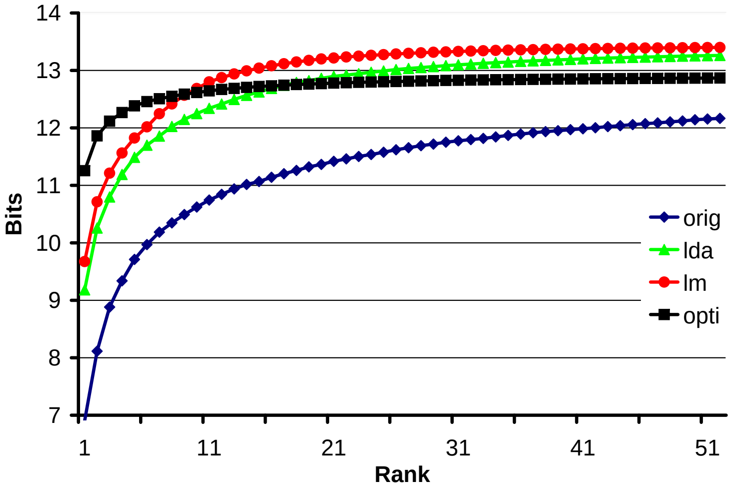
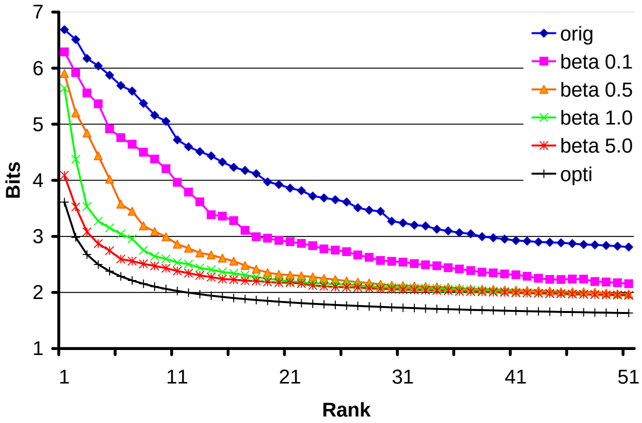


Figure 5.4: Entropy $H(Q)$

The Kullback-Leibler divergence only measures the additional bits needed for representing a distribution Q given an optimal code for the distribution U, i.e., it explicitly disregards the entropy $H(Q)$. Figure 5.4 shows the entropy for the various rerankings. As is to be expected, reranking by minimizing the variance leads to a higher entropy $H(Q)$ at all ranks. Also the optimal ranking *opti* leads to a higher entropy, but levels out at a slightly lower entropy than for the diversified ranks. Naturally, the increased entropy $H(Q)$ also leads to an increased cross-entropy $H(Q;U)$ (not shown). One consequence of this is that the improvement in divergence by diversification is less pronounced for $D_{KL}(Q||U)$ (see Figure 5.2) than for $D_{KL}(U||Q)$. After about rank 15, the gain of diversification is balanced by the cost of diversification in terms of entropy.

The effects of diversification for Yahoo! search results are similar; see last paragraph in Section 5.2.3 for a comparison of Yahoo! and Google.



β	**LDA**	**LM**
0.1	0.56	0.29
0.5	0.33	0.10
1.0	0.27	0.07
5.0	0.16	0.03
Opt	0.24	0.24

Figure 5.5: Kullback-Leibler divergence $D_{KL}(U||Q)$ for different β using language model reranking (left) and Spearman's rank correlation coefficient for different β and for the optimal ranking

Balancing Relevance and Diversity In this section we analyse the effect of the parameter β, which balances between relevance and diversity. To this end, we selected 10 queries, where the difference between divergence of the top 10 results and of the complete result is maximal and varied β between 0.1 and 5. Figure 5.5 (left) compares the divergence using language models as document representations and shows how the Kullback-Leibler divergence of the rerankings with different βs lie in between the original ranking and the optimal ranking. Increasing β beyond 5.0 does not further improve the results.

The right table in Figure 5.5 compares the rank correlation for both search engines and for the optimal ranking. The general behaviour is consistent. The divergence decreases at all ranks with increasing β at the cost of a higher degree of reranking, resulting in a lower rank correlation ρ. $\beta > 1$ achieves only a relatively small improvement, $\beta > 5$ (not shown) achieves no further visible improvement. As already observed in Section 5.2.3, diversification based on latent topic models *lda* generally achieves a ranking closer to the original ranking than diversification based on language models *lm*.

Comparing Two Search Engines Search engines certainly also make an effort towards covering the most important aspects of queries as one of their optimization objectives. Our evaluation framework can also be used to compare topic coverage in the top-k results for different search engines. Figure 5.6 shows the difference $D_{KL}(\text{Google}) - D_{KL}(\text{Yahoo!})$ of the two evaluated search engines of the symmetric Kullback-Leibler divergence:

$$D_{KL}(U,Q) = \frac{D_{KL}(U||Q) + D_{KL}(Q||U)}{2} \tag{5.13}$$

One graph shows the divergence with respect to Wikipedia and the other one with respect to the complete search result. Apparently Google search results tend to be significantly less diverse than Yahoo!'s search results; in the top ranking positions the divergence of Google is almost 1 bit higher than the divergence of Yahoo!.

Of course such a comparison should not be taken as evidence on any inherent bias of a search engine. Firstly, the observable difference may in part be due to different strategies of including Wikipedia pages, which were discarded for evaluation. Secondly, the two search engines employ slightly different strategies in grouping related search results, which were

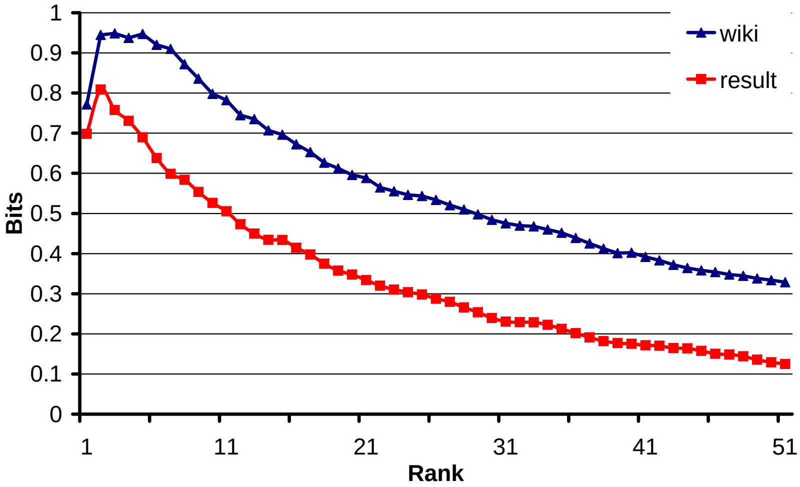


Figure 5.6: Comparison of the diversity of Google and Yahoo! using symmetric $D_{KL}(\text{Google}) - D_{KL}(\text{Yahoo!})$

not taken into account in our evaluation, where we mapped search results to a flat ranked list. Finally, of course Wikipedia does not necessarily cover all possible interpretations of a query.

5.2.4 Results for Evaluation Method

To verify the viability of our proposed diversity evaluation based on Kullback-Leibler divergence and Wikipedia, we compare the results with two other diversity evaluation frameworks: Subtopic retrieval from TREC and a manual evaluation using hand annotated search results from Santamaría et al. [164].

Comparison with TREC Evaluation In order to assess our proposed evaluation criterion based on Wikipedia coverage, we have applied our diversification approach on TREC data. In the Web Track 2009, TREC introduced a dataset to evaluate subtopic coverage of rankings [39]. They provided a Web crawl, 50 queries, and automatically extracted subtopics for these queries. This extraction was done using the query log of a commercial search engine, co-click data, and other information. A set of Webpages from the crawl was then annotated manually with the relevant subtopic or with “not relevant” in case the page does not cover any subtopic.

To compare this evaluation framework with our Wikipedia-based approach we identified a subset of the data satisfying our requirements:

1. A query in Wikipedia must return at least 100 Wikipedia pages.

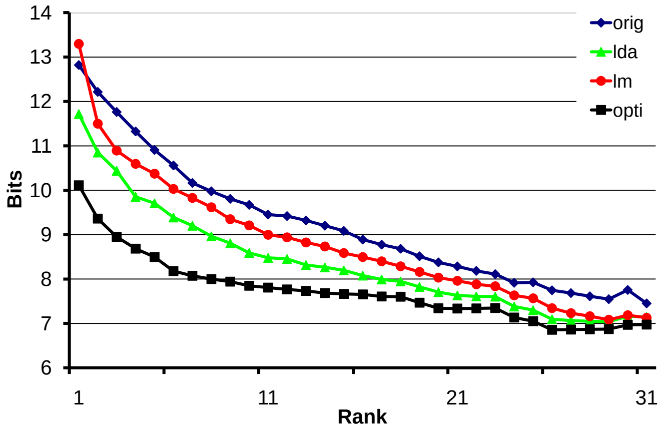
2. An annotated document must occur in the top 1000 results of Google.

Among the 50 TREC queries, 7 did not yield any Wikipedia page, 8 less then 10, and 8 less then 100 result pages when searching Wikipedia. This leaves us with 27 queries with up to 500 ranked, relevant Wikipedia pages. The average overlap of Web search results from our crawl with annotated TREC pages matched by URL is 26.6 pages per query for Google. This leaves us with only a few pages annotated as relevant for a specific subtopic and many queries with no annotated pages for a certain subtopic.

Figure 5.7 (left) shows how the original ranking, and the introduced reranking approaches approximate the corresponding Wikipedia content for the 27 TREC queries (c.f. Figure 5.1). Again, by construction, the optimal ranking covers Wikipedia content best in the top k results. However, for the TREC queries, diversification based on the LDA topics slightly outperforms diversification based on the language models. The original ranking depicts the largest distance to Wikipedia.

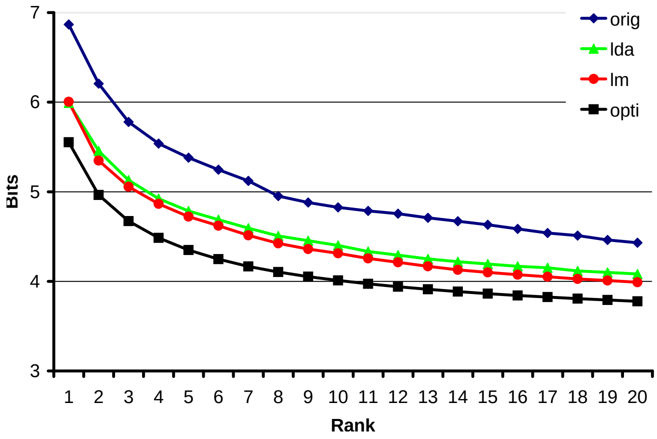
Using the manually assessed subtopics and evaluation scripts provided by the TREC organizers, we computed α–nDCG [38], shown in Table 5.7 (right). On this small dataset the original ranking and the rerankings by means of minimizing variance perform rather similarly. The slight differences are not statistically significant. Only the “optimal” reranking achieves a significant improvement for all metrics but α–nDCG@5 according to a 2-tailed paired t-test with confidence well above 95% (marked with asterisks in the table). This indicates that diversification based on a more or less representative goal model, such as Wikipedia, can outperform diversification based on analyzing only the search results. Investigating and evaluating such a goal-driven approach to diversification in more detail is an interesting subject for future work.

In summary, for the subset of the TREC queries where we had enough data, diversification based on latent topic models generally achieves better coverage of Wikipedia than diversification based on language models. Probably due to the rather small overlap



	@ **5**	@ **10**	@ **20**
Orig	0.214	0.254	0.306
LDA	0.216	0.253	0.298
LM	0.205	0.255	0.299
Opt	0.254	0.290*	0.329*

Figure 5.7: Kullback-Leibler divergence $D_{KL}(U||Q)$ (left) and α–nDCG values (right) for the TREC evaluation



	@ **5**	@ **10**	@ **20**
Orig	0.690	0.672	0.698
LDA	0.719	0.685	0.695
LM	0.712	0.676	0.695
Opt	0.746	0.707	0.730

Figure 5.8: Comparison of $D_{KL}(U||Q)$ values (left) and α–nDCG values (right) for the manual evaluation

between manual TREC assessments and the search engine results, the original rankings and rerankings achieved similar performance with respect to α-nDCG.

To put this into perspective, we note that the clearly best run in the diversity task of TREC 2009 [39] also just took the original ranking provided by a major commercial search engine. Thus the achieved improvement over the original ranking is fairly remarkable.

Comparison with Manual Evaluation As a second dataset to validate our evaluation method we used a test corpus compiled by Santamaría et al. [164]. This corpus comprises Web search results for 40 ambiguous queries consisting of 15 ambiguous nouns from the Senseval-3 dataset and 25 additional ambiguous nouns, where one of the senses is a band name. For all senses there exists a corresponding Wikipedia article. For each query the top 150 documents have been manually annotated with one or more senses. Documents with little text, disambiguation pages, and documents not corresponding to any Wikipedia sense have been discarded.

On the basis of the manual annotations, we have again computed α–nDCG. Figure 5.8 (right) compares the averaged α–nDCG for the 40 queries with our proposed evaluation criterion of Wikipedia coverage measured by the Kullback-Leibler Distance between the search result and the language model of Wikipedia articles (Figure 5.8 (left)). As can be seen, the relative performance of the various rerankings is the same for both evaluation measures, in particular at smaller ranks. The original ranking *orig* is outperformed by the diversified ranking based on language models *lm* and topic models *lda*, which in turn are outperformed by the optimal ranking *opti* based on Wikipedia. This indicates that our proposed evaluation criterion for diversification, which does not require manual annotation, corresponds well with the widely used measure α–nDCG based on manual annotations. Moreover, the fact that the optimal reranking achieves the best α–nDCG confirmes the observation of Santamaría et al. [164] that Wikipedia can be effectively used as a target model for diversification, provided that it covers the most prominent aspects of a query.

5.3 Topic Models for Product Review Ranking Diversification

In this section we investigate the use of topic models to rank product reviews. The main idea is to use Kullback-Leibler divergence to find an optimal ranking covering all aspects/opinions of a product within the top ranked reviews.

5.3.1 Approach

In contrast to Web search results, reviews for a product can not be ranked based on relevance since all reviews are usually relevant for the product the review is about. Review recommendation or classification is a well-studied problem, but they don't optimize a ranking of reviews but evaluate the reviews individually, for example by letting other users rate a review. We try not to find the best or most helpful individual reviews for a product but to find the top-k reviews to provide the user with a good summary of the opinions about a product. To this end we model the reviews using latent topics extracted with latent Dirichlet allocation (LDA) and the assigned star ratings for the product. The ranking of the reviews is based on Kullback-Leibler divergence (KLD) to get an optimal summary of all reviews for a product with the largest possible topical diversity. Our framework also allows to set a different goal when computing the optimal ranking, e.g. cover all positive aspects of a product, or cover all sentiments associated with an aspect/feature of the product. In the following paragraph we describe the conceptional architecture of the framework.

System Overview. Our framework to diversify review rankings consists of two main components to model the data and to rerank the result lists:

1. The LDA component to model the review data
2. The Ranking component to optimize the ranking based on different strategies

An overview of the system can be seen in Figure 5.9. The input for the system are all reviews written for a product together with the rating assigned by the user to the product. Our hypothesis is that users who assign 5 stars (on a five point liker scale) mainly talk about positive experience with the product or it's features. A review accompanied by a 1 star rating indicates a review with rather negative points. To not exclude the possibility that also in a 5 star review a minor negative point could be expressed we use a matrix allowing to smooth the assignment of reviews to rating classes. Especially a 3 star rating can contain negative as well as positive aspects which can be modeled using this matrix. Based on the topic models for each review we then rank the reviews by minimizing the Kullback-Leibler divergence between the aggregated reviews of the ranking and three other distributions depending on the optimization strategy. After discussing the preprocessing step in the next paragraph, we describe the modeling approach followed by a description of the ranking process.

Preprocessing. Since reviews are user-generated, they contain more grammatical errors, sloppy language, and spelling errors than more carefully written texts. Therefore preprocessing the raw data becomes an important task. We used the Stanford POS Tagger [181] for tokenization and part-of-speech tagging. Then WordNet [58] was used to get the lemmas of the terms and remove all terms that are not verbs, adverbs, nouns, or adjectives.

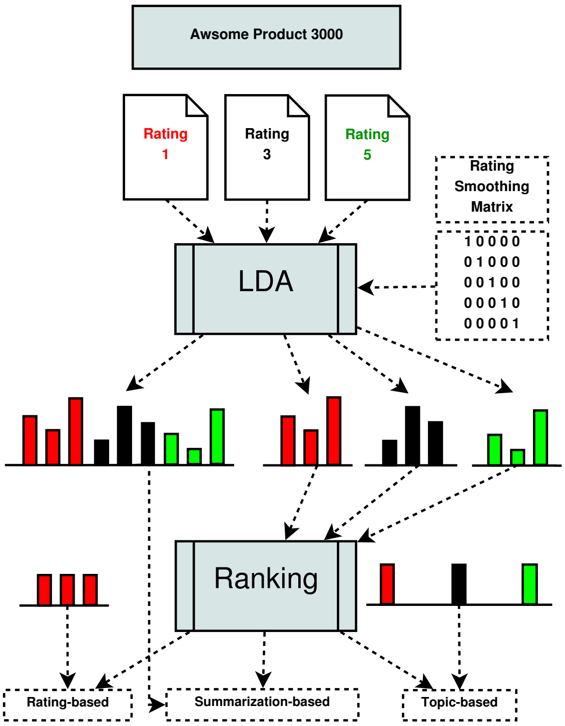


Figure 5.9: Overview of the review ranking system

Since unigrams might not give an accurate picture of what a review is about we extract n-grams of variable lengths in the next step. Especially in the context of product reviews, multi-term phrases are important to model the data, e.g. “Microsoft Windows 7 Professional”, “not recommended”, or “graphic card” (see Section 2.2).

As a result, all documents in our corpus of product reviews are segmented into variable-length n-grams and the latent topics can now be based on n-grams or phrases instead of fixed sized units or single terms. Figure 5.10 shows a snippet of an original product review together with the preprocessed version without stop words but with part-of-speech information and some multigrams.

Modeling Reviews. To model the review data we make use of probabilistic topic models [172] to extract latent topics within the review corpus. We combine this information with the assigned star ratings for the reviews to cover positive and negative statements associated with a particular latent topic. In the following we explain how we combine the star ratings and the information about the extracted latent topics.

Finding Latent Topics. A product review usually covers different aspects or features of a product. For example, users have an opinion about the price of a product or the service of a company. Instead of a fine-grained extraction of features and sentiment, following for instance Bross and Ehrig’s approach [28], we rely on a statistical approach to find features or aspects.

With LDA at hand, we are able to represent latent topics as a list of multigrams (see Section 2.2) with a probability for each multigram indicating the membership degree within the topic. Furthermore, for each document in our corpus (reviews in our case) we can determine to which topics it belongs, also associated with a degree of membership (topic probability $P(z_j \mid d_i)$).

An example for two extracted latent topics represented by the top 10 terms is shown in

. . . Wish burger - also known as a veggie burger (no meat) Ketchup and Mustard are actually available at In and Out.. just ask, its really easy Double Meat - is la double double with no cheese Flying Dutchman. . .

wish.n burger.n also.r know.v veggie.n_burger.n meat.n actually.r available.a ketchup.n mustard.n just.r ask.v really.r easy.a double.r_meat.n double.a_double.a cheese.n flying.n_dutchman.n

Figure 5.10: Preprocessed review snippet from Epinions.com: original on top; Segmented and POS-tagged on bottom

Table 5.4: Top terms composing the latent topics "ticket" and "waiting" for Epinions.com reviews about America West Airlines

"Ticket"-Topic		**"Waiting"-Topic**	
Term	**Prob.**	**Term**	**Prob.**
ticket.n	0.038	concourse.n	0.015
voucher.n	0.027	miss.v	0.015
clerk.n	0.016	take.v_off.r	0.015
care.v	0.011	hour.n_late.r	0.012
availability.n	0.008	change.n	0.009
complain.v	0.008	delay.n	0.009
look.v	0.008	flight.n_attendant.n	0.009
nightmare.n	0.008	meeting.n	0.009
suggest.v	0.008	not.r	0.009
america.n_worst.r	0.006	reggie.n	0.009

Table 5.4. Beside the terms also the probability for the terms belonging to the topic are shown. For this example we used $|Z| = 50$ latent topics.

Combining Latent Topics and Star Ratings. Each review d can now be modeled as a mixture of latent topics $P(z_i \mid d)$. Together with the rating of each review $r(d)$ we can transform the topic model into a topic-rating model by considering the topics for each rating class $r \in R = \{1, \ldots, 5\}$ separately:

$$P(z'_k \mid d) = \sum_{r \in R} m_{r(d)-1,r-1} * P(z_{k\%|Z|} \mid d), \tag{5.14}$$

where $k = \{0, \ldots, |R| * |Z|\}$ and $m_{i,j}$ an entry in the rating smoothing matrix:

$$M = \begin{pmatrix} 0.6 & 0.3 & 0.1 & 0.0 & 0.0 \\ 0.4 & 0.5 & 0.1 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.6 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.5 & 0.4 \\ 0.0 & 0.0 & 0.1 & 0.3 & 0.6 \end{pmatrix} \tag{5.15}$$

The matrix defines how likely it is that, e.g. a negative review contains neutral or positive aspects. This is also dependent on the dataset and the typical user rating behavior.

We also experimented with other matrices yielding similar results. An optimization of this matrix for different rating sites or even different products might improve results further.

All latent topics extracted by LDA are now represented individually for each rating class. Each review is modeled as a topic mixture depending on its rating with some overlap according to the rating smoothing matrix. In the next section we describe how to compute the reference topic models to compute the different rankings corresponding to various strategies.

Ranking Reviews. Depending on the user's information need we define three ranking strategies:

1. Summary-focused Ranking
2. Sentiment-focused Ranking
3. Topic-focused Ranking

To compute these rankings we take the topic-rating models of the reviews computed in the previous step and try to minimize the distance between the aggregated top-k reviews and a strategy-dependent target distribution. We use a greedy algorithm to find the best review for each position in the ranking.

As a measure for how well the top-k reviews approximate the corresponding target distribution we calculate the Kullback-Leibler divergence between the smoothed topic-rating models for the top-k results and for the target distributions. Kullback-Leibler divergence estimates the number of additional bits needed to encode the distribution U, using an optimal code for Q, and having a combined vocabulary size of $|Z'|$; in our case the number of latent topics $|Z|$ times the number of rating classes $|R|$.

$$D_{KL}(U||Q) = H(U;Q) - H(U) = \sum_{i=1}^{|Z'|} u_i * log_2(\frac{u_i}{q_i}) \tag{5.16}$$

In our setting, distribution Q is the combined topic-rating model of the top-k reviews and thus $D_{KL}(U||Q)$ can be directly used to measure the similarity with the target distribution.

Summary-focused Ranking. In most cases users reading reviews are interested in getting an overview of the experiences of other users with the product. A ranking which gives a good overview summarizes the views expressed in all reviews. The goal for a review ranking system is therefore to approximate all reviews by the top-k in the ranking. Thus, the top-k reviews *summarize* the opinions about a product present in all reviews.

With the topic-rating models computed for each review we try to find a ranking of reviews that approximates the aggregated topic-rating models of all reviews for a product. This means we try to minimize the Kullback-Leibler divergence between the top-k ranked reviews and the aggregation of all reviews.

To clarify the functioning of the greedy algorithm let's consider a product with three reviews A, B, and C. We compute the aggregated topic-rating model $A + B + C$ and measure the Kullback-Leibler divergence D_{KL} for each position in the ranking. The example is shown in Figure 5.11.

Sentiment-focused Ranking. Instead of approximating all reviews, the sentiment-focused ranking tries to summarize only one particular class of ratings, for example negative aspects as represented by the topic-rating model with rating one. It could also be interesting to see which features of a product are discussed mainly in a neutral review. Because of the rating smoothing matrix also aspects from reviews having a different rating influence the ranking.

1	D_{KL}	A	B	C
	$A+B+C$	0.5	0.3	0.7
			Rank 1	
2	D_{KL}	$B+A$	–	$B+C$
	$A+B+C$	0.2	–	0.3
		Rank 2	Rank 1	
3	D_{KL}	–	–	$A+B+C$
	$A+B+C$	–	–	0.0
		Rank 2	Rank 1	Rank 3

Figure 5.11: Example of the greedy algorithm to find a ranking summarizing the three reviews A,B, and C

The target distribution that we try to approximate with the review ranking in this case is a (smoothed) uniform distribution over all topics for one rating. That means we get a diverse ranking covering all latent topics associated with a particular rating.

Topic-focused Ranking. Analog to the previous sentiment-focused ranking we can focus the review ranking on a particular latent topic. This allows to get all opinions – positive, neutral, and negative – about a certain aspect. This might be useful for users who are interested in a particular feature of a product and the experience other users report in their reviews.

This type of ranking can be achieved by minimizing the Kullback-Leibler divergence of the reviews in the ranking and a (smoothed) uniform target distribution over all ratings for one topic. We refrained from evaluating this strategy due to the missing of large scale user data for this type of task and the difficult mapping of latent topics to well-defined product features.

5.3.2 Evaluation

To evaluate our system and the different rankings we adopt a method from information retrieval to judge rankings based on novelty and diversity. The ideal ranking would cover all different aspects and all different opinions about the aspects. The first review in the ranking should cover many aspects of the product to serve as a good overview. This can be compared to sub-topic retrieval where Web search engines try to find an optimal ranking to cover as many sub-topics as possible (see, for example, TREC 2009 Web Track, Diversity Task [39]). This evaluation approach requires annotated results, namely each review needs to be annotated with the sub-topics discussed in it. In the following we describe our dataset and the annotation of the test data.

Dataset. We crawled the Epinions[13] Web site to get for each of 300 products around 100 reviews. Since we followed a user-focused approach to crawl the data, we got many more reviews then needed. For the evaluation we needed to manually annotate the reviews with features of the product covered by the review and the polarity. Out of the 300 products we randomly picked two having not only positive or negative ratings: "America West Airline" and "Pokemon Snap for Nintendo 64". Table 5.5 shows the distribution of ratings for these two products in our corpus.

For manually annotating 200 reviews we first identified different features of the products. Table 5.6 shows the annotation form to annotate the reviews for "America West Airlines".

[13] www.epinions.com

Table 5.5: Distribution of the ratings for the test products

Rating	Number of Posts for	
	"Pokemon"	"America West"
1.0	13	43
2.0	23	29
3.0	22	23
4.0	32	17
5.0	13	5

Table 5.6: Sample annotation form for "America West Airlines" reviews

Feature/Aspect	Positive	Negative
Pricing	X	
Seating/space		
Food		X
Customer service		
Compliance with timetable		
Gate changes		
Luggage condition		X
Frequent flyer program	X	
General aspects		X

5.3.3 Results

To evaluate the summarization-based ranking we computed α-nDCG [38] for our rankings using the manually annotated reviews to assess relevance and novelty. The results for the top-20 reviews using different numbers of latent topics and $\alpha = 0.5$ are shown in Figures 5.12 and 5.13. The best performance for "America Airline West" is achieved using 25 latent topics whereas for "Pokemon" 10 topics are the best choice. This can be explained by having a closer look at the individual reviews. The "Pokemon" reviews are considerably shorter and have clearly defined features. The airline has more features, longer reviews and are written in a more narrative style, e.g. "I had a 10 day vacation flying out of DC National on December 29, 2000 and spending 3 days in. . . ". The "Pokemon" reviews on the other hand are more to the point: E.g. "This game is boring." or "This game is great in the beginning.". Figure 5.13 also shows that for "Pokemon" reading three reviews is already enough to get a good overview of the different opinions. For "America West", the user has to go through the first 8 reviews to cover most of the aspects and opinions. The general problem of LDA to find the best number of latent topics is also an issue for review ranking. Predicting this number for each product is an interesting task for future work.

The influence of smoothing the rating by assigning a fuzzy membership degree for each review to the review classes is shown in Figures 5.14 and 5.15. "With smoothing" indicates the use of the rating smoothing matrix as depicted in Equation 5.15. "No smoothing" means that the different rating classes are strictly kept separately with a diagonal rating smoothing matrix $M = (m_{i,j})$ with $m_{i,j} = 1.0$ if $i = j$ and else $m_{i,j} = 0.0$. The results for using a rating smoothing matrix $M = (m_{i,j})$ with $m_{i,j} = 0.2, \forall i, j \in \{1, \ldots, |R|\}$ is labeled "Equal Smoothing". Different variations of the rating smoothing matrix could be necessary for different datasets depending for example on the skewness of the rating distribution over the classes or on individual user preferences.

The results for sentiment-focused ranking using 10 latent topics and focusing on either positive or negative aspects are shown in Figure 5.16. To compute the α-nDCG values

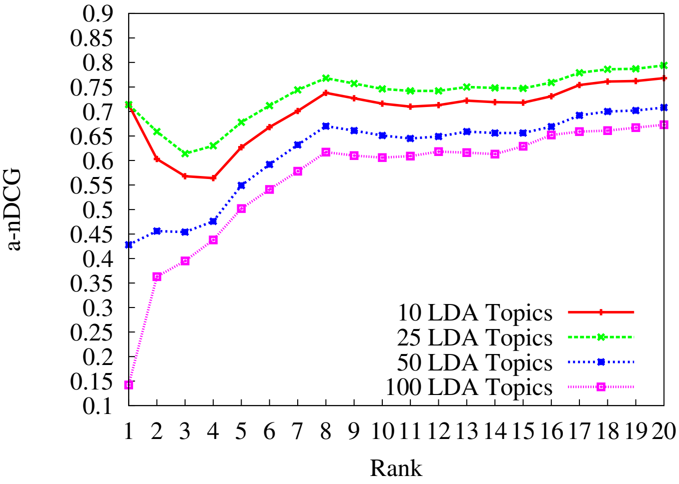


Figure 5.12: Results for different number of latent topics for “America West Airlines”

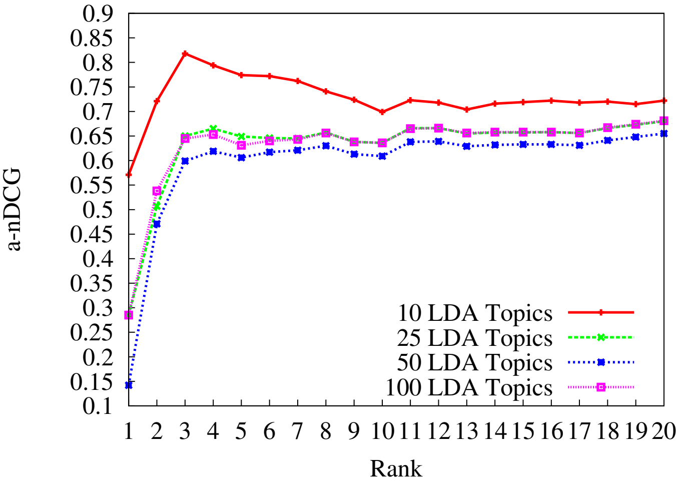


Figure 5.13: Results for different numbers of latent topics for “Pokemon”

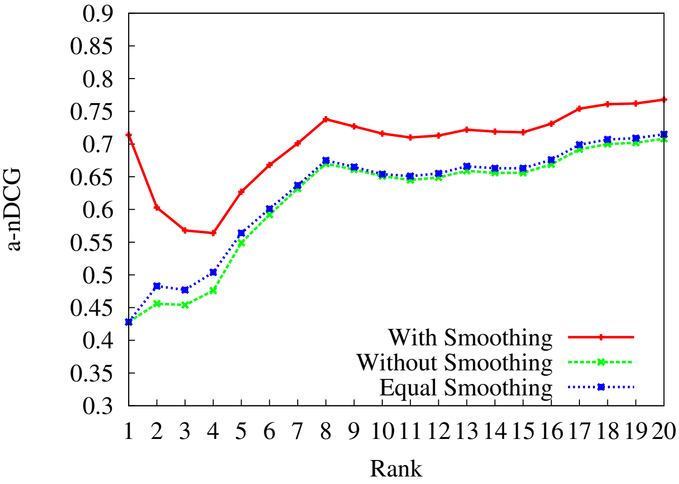


Figure 5.14: Influence of the rating smoothing matrix for “America West Airlines”

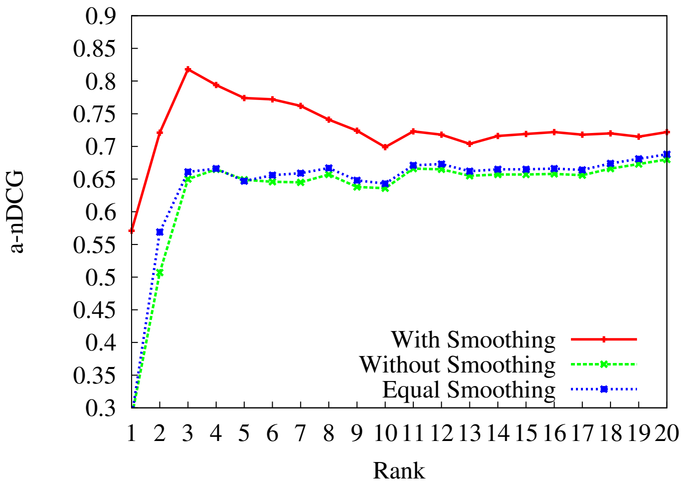


Figure 5.15: Influence of the rating smoothing matrix for "Pokemon"

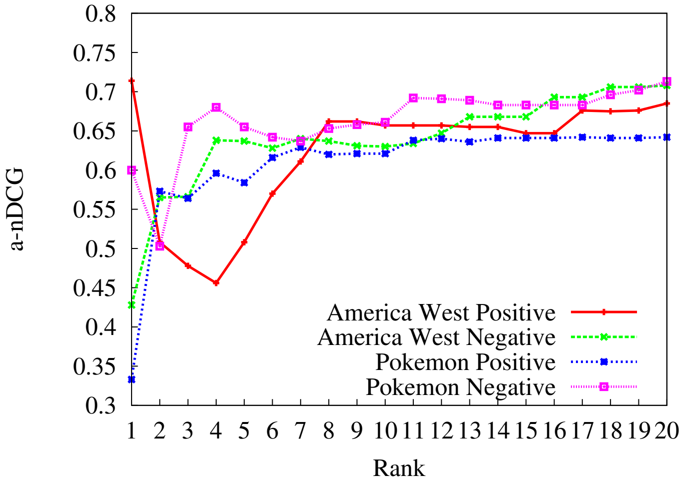


Figure 5.16: Results for sentiment-focused ranking summarizing only positive or negative aspects respectively

we only considered the positive, respectively negative, manually annotated aspects to be relevant. As can be seen in the figure, summarizing the negative opinions with the top-k reviews in the ranking for "Pokemon' is easier than the positive opinions. For "America West" the negative aspects are quite well covered after the first four reviews whereas the coverage of positive opinions is at its minimum at this position in the ranking.

5.4 Related Work

Search result diversification has received considerable attention in the past years; for recent overviews on the main issues and current approaches see, for example, [153].

Diversifying Result Lists. One of the first works on result diversification introduces Maximum Marginal Relevance (MMR) as a ranking measure that balances relevance as the similarity between query and search results with diversity as the dissimilarity among search results [30]. Notably, MMR has not only been successfully used for diversity aware ranking, but also for text summarization, by selecting relevant and diverse text passages that cover the main topics or aspects of a text. Top-k diversification as pursued in our approach has the similar goal of covering the main aspects of a query by the top ranked search results.

Other approaches like [213] diversify recommendation lists to accommodate a users full spectrum of interests and minimize redundancy among the recommended items. Reranking approaches to diversify search results, e.g. [154] is based on query reformulations obtained from a query log where the focus lies on personalized search results, or [34] who describe a Bayesian reranking approach to maximize the coverage of different meanings of a query in the top 10 results have been explored before. Zhai and Lafferty [210] use statistical models for queries and documents. They model user preferences as loss functions and the retrieval process as a risk minimization problem. They retrieve models for subtopic retrieval that take dependencies between search results into account.

More recent approaches to diversification all essentially balance relevance with diversity, but differ in estimation of relevance and similarity, and choice of diversification objective. Agrawal et al. [2] classifies queries and results to categories of the ODP taxonomy, and diversifies results by maximizing the sum of categories covered by the top-k results weighted by the probability of categories given the query. Thereby the risk that the top-k results contain no relevant result for some category at all is minimized. Gollapudi and Sharma [69] introduce a framework for analyzing approaches to diversification as variants of facility dispersion. On this basis they analyze and evaluate three diversification objectives: MaxSum, which takes into account all pairwise dissimilarities between top-k results as a measure for diversity, MaxMin, which maximizes just the minimum relevance and dissimilarity of results, and MonoObjective as a weighted aggregation of relevance and average dissimilarity for each top-k result. Wang and Zhu [188] introduces an approach to search result diversification adopting the Modern Portfolio Theory of finance. They generalize the well-known probability ranking principle (PRP) [160] by maximizing not only the relevance of top-k results but also minimizing the (co-)variance of the results. A greedy algorithm is used for ranking search results such that relevance is maximized while variance is minimized. Rafiei et al. [156] introduces a similar framework based on Portfolio Theory for reranking Web search results. Other than the greedy algorithm used in [188] they use quadratic programming optimization for arriving at optimal portfolios.

Santamaria et al. [164] investigated the use of Wikipedia to improve diversity in Web search results. They manually annotated the top 100 results of a Web search engine for a set of 40 nouns with Wikipedia senses extracted from disambiguation pages. They showed that Wikipedia senses cover 56% of the Web pages and thus Wikipedia is much more suited than other sense inventories like WordNet (32%). Additionally, they propose using a vector space model and cosine similarity or word sense disambiguation algorithms to assign a Wikipedia sense to each page. Maximizing the number of different Wikipedia senses is then the goal of their greedy reranking algorithm.

The problem of result diversification is also investigated in the area of structured data queries. Recommending a set of items to the user or returning a list of products in response to a keyword query are applications for result diversification. Vee et al. [185] propose an efficient algorithm to find a representative, diverse set of top-k results for a given form-based query. All attributes of an object are ordered according to their priority for diversification by a domain expert. Jain et al. [91] make use of k-nearest neighbor clustering techniques and combine it with a notion of diversity based on a distance metric. Each query is represented as a point in an n-dimensional space and the k-nearest neighbors are selected which also satisfy the required distance. Demidova et al. [52] go a step further by introducing an approach that diversifies keyword queries against structured databases based on their schema rather than diversifying the results. A necessary condition for these approaches is that the database schema captures the semantics of the domain at hand.

We follow the approach of Wang and Zhu [188] to minimize (co-)variance for maximizing diversity, but rely on the search engine ranking for estimating relevance rather than estimating it from the documents directly. Moreover, we evaluate to what extent a condensed representation in terms of latent topic models can capture diversity better than the language modeling approach used in [188].

Diversity Evaluation. Evaluation of result diversification requires new measures that consider more than just simple relevance judgements. To this end, several extensions to traditional measures have been proposed. Their common idea is that queries and documents cover several subtopics (also called aspects or nuggets), and thus relevance is assessed w.r.t. subtopics rather than w.r.t. documents.

Chen and Karger [34] evaluate their approach on different TREC tasks (robust track, interactive track, and manually annotated TREC data). In [154] user assessment of the result is used to measure whether the diversified result list contains at least one document satisfying the user's interest. Zhai et al. [207] introduce variants of recall and precision that take into account the subtopics of a query. S-recall at K measures the proportion of subtopics covered by the top-k results, and S-precision at r measures the ratio of the best rank K_{opt} that can be achieved for a given recall r and the actual rank K with recall r.

Clarke et al. [38] introduce α–nDCG as a generalization of the nDCG measure (normalized Discounted Cumulative Gain). Whereas nDCG only measures the relevance of search results, discounted by the logarithm of their rank, α–nDCG in addition penalizes repeated subtopics in search results. For evaluating diversification of search engine results the required explicit relevance assessments in terms of subtopics are difficult to acquire. Gollapudi and Sharma [69] avoid the need for human relevance assessment by taking Wikipedia pages returned for a query as subtopics, and estimate subtopic relevance by a thresholded similarity between result documents and Wikipedia pages to measure S-recall (also called novelty). We also compare original and diversified rankings with respect to Wikipedia, but estimate "subtopic coverage" directly on the language models of the top-k results and Wikipedia.

Review Recommendation and Summarization. Generally review recommendation techniques can be seen as an explanation [177, 178] or classification problem [138]. O'Mahony et al. [138] give an overview over existing machine learning methods for review recommendation. Kim et al. [99] use SVM regression on structural, lexical, syntactic, semantic, and sentiment features to classify reviews, and stated that helpfulness is very dependent on the length of a review, its unigrams and score. Liu et al. [123] have shown that helpfulness of movie reviews are expertise and time dependent. While the majority of existing work utilize text categorization techniques for recommending reviews Harper et al. [76] train their classifier according to features relating to question categories, text categorization and social networking metrics. Credibility assessment has also been considered by Weekamp et al. [189] to consider features such as timeliness of posts, post length and spelling quality in topical reviews.

Existing work on review summarization falls under subjective classification [194], sentiment analysis [33], or under traditional text summarization. Classic text summarization methods can be categorized into two: template instantiation [96] and passage extraction [162]. Researchers differentiate between review summarization and classic text summarization techniques [212]. Sentiment analysis techniques try to produce a summarized sentiment consisting of sentences from a source document presenting the opinion and idea of it's corresponding author. With respect to length and structure, this summary can be either a single paragraph [13] by careful selection of sentences or the source document, or a structured sentence [85], which is in turn generated by mining features that the author has commented on. To build summaries of sentence list structures, Hu and Liu [85] introduced a method utilizing word attributes such as frequency of occurrence, part-of-speech tagging and WordNet synsets. Following this approach features are extracted, combined with their contextually close words, and finally used to generate a summary by selecting and restructuring the sentences following the extracted features. Implementations following text sentiment analysis have been proposed such as Opine [148], which uses relaxation labeling to find the lexico-semantic orientation of words, or Pulse [61] which uses bootstrapping to train a sentiment classifier using features extracted by labeling sentence clusters with respect to their key terms.

In comparison with these works we summarized reviews by choosing complementing reviews and ranked them according to different strategies. The product ratings served as an indicator for the sentiment, and the extracted latent topics ensured topical coverage of relevant aspects. Therefore we proposed a greedy algorithm to minimize the Kullback-Leibler divergence (KLD) between the topic models of the top-k results and all reviews for a product. KLD has, for example, been used as a similarity measure for audio files [166], while we used it to diversify topic models.

Our Contributions. We have presented a reranking approach for balancing the top-k results of Web search engines with respect to diversity by minimizing the variance of their underlying language models and topic models. Our extensive evaluation against Wikipedia has demonstrated that the approach effectively achieves a better coverage of the various topics and aspects pertaining to a query. We further demonstrated that diversification based on language models achieves a slightly better coverage in terms of Wikipedia language models than diversification based on topic models, but topic models accomplish diversification with a significantly lesser amount of reranking. Our evaluation using the TREC data and supplied evaluation framework confirms these findings and validates the presented Wikipedia-based diversity evaluation as an alternative to costly manual diversity assessment.

Additionally, we presented an approach to rank reviews for products based on latent topics and user-assigned ratings. The main goal was to summarize the opinions expressed in all reviews for a product in the top-k results of a ranking. In contrast to recommending single reviews we aimed at recommending an optimal diverse set of reviews using LDA and KLD. Manual annotation of the reviews allowed an automatic evaluation of the proposed approach comparing different ranking strategies and demonstrating the effectiveness of the approach.

CHAPTER

SIX

TAG RECOMMENDATION

In this chapter we investigate the use of topic models to do collective tag recommendation independent of individual users. We further investigate the use of language models to personal tag recommendation and show how the combination of both improves over state-of-the-art systems.

6.1 Introduction

Tagging systems [129] like Flickr[1], Last.fm[2] or Delicious[3] have become major infrastructures on the Web. These systems allow users to create and manage tags to annotate and categorize content. In *social* tagging systems like Delicious the user can not only annotate his own content but also content of others. The service offered by these systems is twofold: They allow users to publish content and to search for content. Thus *tagging* also serves two purposes for the user:

1. Tags help to organize and manage own content, and
2. Find relevant content shared by other users.

Tag recommendation can focus on one of the two aspects. Personalized tag recommendation helps individual users to annotate their content in order to manage and retrieve their own resources. Collective tag recommendation aims at making resources more visible to other users by recommending tags that facilitate browsing and search. However, since

[1] `http://www.flickr.com`
[2] `http://www.lastfm.com`
[3] `http://delicious.com`

tags are not restricted to a certain vocabulary, users can pick any tags they like to describe resources. Thus, these tags can be inconsistent and idiosyncratic, both due to users' personal terminology as well as due to the different purposes tags fulfill [68]. This reduces the usefulness of tags in particular for resources annotated by only a few users (aka cold start problem in tagging), whereas for popular resources collaborative tagging typically saturates at some point, i.e., the rate of new descriptive tags quickly decreases with the number of users annotating a resource [81].

In Section 6.2 we present an approach to overcome the cold start problem for tagging new resources. To this end, we use latent Dirichlet allocation (LDA) to elicit latent topics from resources with a fairly stable and complete tag set to recommend topics for new resources with only a few tags. Based on this, other tags belonging to the recommended topics can be recommended. Compared to an approach using association rules, suggested previously for tag recommendation, our approach achieves significantly better precision and recall. Moreover, the recommended tags are more specific for a particular resource, and thus more useful for searching and recommending resources to other users [19].

Tagging is considered a categorization process not a classification process, see [75]. The underlying meaning has to be evaluated and inferred in the context of other tags and user information. Tags can even have no concrete meaning or are only interpretable by the user herself. In addition, tags can have various purposes. Some describe the annotated content, some refer to the user (e.g. "jazz", "myHolidays" or "to_read") as described in [19]. In practice allowing users to freely annotate means that tagging systems contain noise and are rather sparsely populated. Studies by [67] and [27] have shown that many users annotating a resource leads to a stable tag distribution for this resource, capturing its characteristics sufficiently. To support users in choosing tags, tag recommendation algorithms have emerged. For resources already annotated by lots of people this recommendation is rather straight forward. The tagging system can provide the most frequent tags assigned to the resource, or look at the tagging history of the user to make a more personalized recommendation. On this note, tag recommendation algorithms can be classified into user-centered and resource-centered ones, see [205].

In Section 6.3 we combine both perspectives to recommend personalized tags to users. To this end, we employ a mixture of simple language models (LM) and latent Dirichlet allocation (LDA) to estimate the probability of new tags based on the already assigned tags of a resource and a user, and introduce a principled approach for combining these estimates. The potential advantage of employing LDA is the possibility to recommend tags not previously assigned to the resource or used by the user. This broadens the available vocabulary for tag recommendation. The potential advantage of combining the resource perspective with the user perspective is to filter general tags for a resource with the individual tagging preferences of a user. We evaluate our approach on two real world datasets. We systematically analyze tag recommendation based on resources or users only, assess the possible merits of LDA as opposed to language models, and compare our combined approach to FolkRank by [84] as a state-of-the-art personalized tag-recommender. Our evaluation shows that combining evidence from the resource and the user improves tag recommendation significantly, and that LDA helps, in particular for generalizing from individual tagging practices on resources. Moreover, our approach achieves significantly better accuracy than state-of-the-art approaches.

Problem Statement and Description. Given a set of resources R, tags T, and users U, the ternary relation $Y \subseteq R \times T \times U$ represents the user specific assignment of tags to resources. A bookmark $b(r_i, u_j)$ for resource $r_i \in R$ and a user $u_j \in U$ comprises all tags assigned by u_j to r_i: $b(r_i, u_j) = \pi_t \sigma_{r_i,u_j} Y$[4]. The goal of collective tag recommendation is to suggest new tags for a resource r_i with only a few bookmarks based on tag assignments to other resources collected in $X = \sigma_{r \neq r_i} \pi_{r,t} Y \subseteq R \times T$. The goal of personalized tag recommendation is to suggest new tags for a resource r_i taking into account other tag assignments by the same user to other resources.

[4] projection π and selection σ operate on multi-sets without removing duplicate tuples

6.2 Topic Models for Tag Recommendation

In this section we present our approach for tag recommendation for resources. In contrast to personalized tag recommendation, this collective tag recommendation should help to overcome the cold start problem in tagging systems. We show that LDA can be employed successfully for resource-centered recommendation. We demonstrate that automatically generated tags using our algorithm improves search in tagging systems. To evaluate our approach we compare it to association rules – a state-of-the-art method for tag recommendation proposed e.g. by Heymann et al. [81].

6.2.1 Approach

In the context of tagging systems where multiple users are annotating resources, the resulting topics reflect a collaborative shared view of the document and the tags of the topics reflect a common vocabulary to describe the document.

LDA assigns to each document latent topics together with a probability value that each topic contributes to the overall document. For tagging systems the documents are resources $r \in R$, and each resource is described by tags $t \in T$ assigned by users $u \in U$. Instead of documents composed of terms, we have resources composed of tags. To build an LDA model we need resources and associated tags previously assigned by users. For each resource r we need some bookmarks $b(r, u_i)$ assigned by users $u_i, i \in \{1 \dots n\}$. Then we can represent each resource in the system not with its actual tags but with the tags from topics discovered by LDA.

For a new resource r_{new} where we only have a small number of bookmarks ($i \in \{1 \dots 5\}$), i.e., only one to five users annotated this resource, we can expand the latent topic representation of this resource with the top tags of each latent topic. To accommodate the fact of some tags being added by multiple users whereas others are only added by one or two users we can use the probabilities that LDA assigns. As formalized in Equation 2.5 this is a two level process. Probabilities are assigned not only to the latent topics for a single resource but also to each tag within a latent topic to indicate the probability of this tag being part of that particular topic. We represent each resource r_i as the probabilities $P(z|r_i)$ for each latent topic $z_j \in Z$. Every topic z_j is represented as the probabilities $P(t|z_j)$ for each tag $t_n \in T$. By combining these two probabilities for each tag for r_{new}, we get a probability value for each tag that can be interpreted similarly as the relative tag frequency of a resource. Setting a threshold allows to adjust the number of recommended tags and emphasis can be shifted from recall to precision.

Imagine a resource with the following tags: “photo”, “photography”, and “howto”. Table 6.1 shows the top terms for two topics related with the assigned tags. It is interesting to compare these two topics with the corresponding association rules in Table 6.2. Whereas the association rules indicate only fairly simple term expansions, the latent topics comprise an arguably broader notion of (digital) photography and the various aspects of tutorial material. Given these topics we can easily extend the current tag set or recommend new tags to users by looking at the latent topics. In our example, we can recommend “photos”, “images”, “photoshop”, “tutorial”, “reference”, and “tips” if we set the threshold for the accumulated probabilities to 0.045 . LDA would assume that our resource in question belongs to 66% to the “photo”-topic and to 33% to the “howto”-topic. Multiplying these probabilities with the individual tag probabilities of the latent topics results in a ranked list of relevant tags for our resource.

Table 6.1: Top terms composing the latent topics “photography” and “howto”

“Photography”-Topic			**“Howto”-Topic**		
Tag	**Count**	**Prob.**	**Tag**	**Count**	**Prob.**
photography	16452	0.235	howto	23371	0.219
photo	9002	0.129	tutorial	15519	0.145
photos	7739	0.110	reference	14084	0.132
images	6302	0.090	tips	13955	0.131
photoshop	4825	0.069	tutorials	7320	0.069
graphics	2831	0.040	guide	3430	0.032
image	2769	0.040	toread	2948	0.028
art	1910	0.027	article	2376	0.022
stock	1852	0.026	articles	1498	0.014
pictures	1676	0.024	useful	1442	0.013
design	1666	0.024	learning	1147	0.011
gallery	1386	0.020	tricks	1140	0.011
camera	831	0.012	how-to	1081	0.010
digital	802	0.011	help	1054	0.010

Association Rules Association rules have been investigated in [81] for tag recommendation. They have the form $T_1 \to T_2$, where T_1 and T_2 are tag sets. The three key measures for association rules are support, confidence, and interest. Support is the (relative) number of resources that contain all tags of T_1 and T_2, i.e., an estimate of the joint probability $P(T_1, T_2)$. Confidence is an estimate of the conditional probability $P(T_2|T_1)$, i.e., how likely is T_2 given T_1. Interest (also called lift) is defined as the ratio between the common support for T_1 and T_2, and the individual support of T_1 and T_2 ($\frac{P(T_1,T_2)}{P(T_1)P(T_2)}$), and indicates whether T_1 and T_2 occur more often together than expected, if they were statistically independent. There exist efficient algorithms to exhaustively mine association rules with some minimum support from large datasets (e.g. [3]).

The basic idea of using association rules for tag recommendation is simple: If many resources with tags T_1 (high support) are typically also annotated with tags T_2 (high confidence), then a new resource with tags T_1 may also be meaningfully annotated with tags T_2. More formally, given the tag set from (a few) bookmarks $T = \bigcup b(r, u_i)$ for a resource r by users u_i, and an association rule $T_1 \to T_2$, all tags in T_2 are recommended, if $T_1 \subseteq T$.

Table 6.2 gives a selection of association rules with high confidence mined from our dataset. As also observed in [81] these rules cover all sorts of terminological relationships including spelling variants and synonyms (humour → humor; tools, utilities, utility → tool), loose notions of hypernyms (tutorial, resources → reference), and closely related terms (software, mac, apple → osx).

While the mined association rules are very intuitive, they typically recommend rather generic, frequent tags, such as “software” or “web”. This is a direct consequence of requiring some minimum support for T_1 and T_2. Such generic tags are not necessarily useful for finding specific resources. Indeed, for personalized tag recommendation Xu et al. [202] explicitly penalize tag co-occurrences, when they have been annotated by different users.

6.2.2 Evaluation

Dataset As a dataset for our evaluations we use a crawl from Delicious provided by Hotho et al. [84]. The dataset consists of ~75,000 users, ~500,000 tags and ~3,200,000 resources connected via ~17,000,000 tag assignments of users.

Table 6.2: Selection of tag association rules with confidence ≥ 0.9

Conf	Supp	Int	Rule
0.978	0.037	10.13	web, js → javascript
0.921	0.012	6.75	software, macintosh → mac
0.919	0.161	1.36	tools, fun, interesting → cool
0.915	0.086	4.05	web, weblogs → blogs
0.914	0.074	7.71	humour → humor
0.912	0.037	11.57	photography, photos → photo
0.910	0.136	3.81	howto, code, tutorials → tutorial
0.904	0.086	2.42	tools, utilities, utility → tool
0.904	0.111	2.55	tech, tutorial, tutorials → howto
0.902	0.049	1.71	toread, howto, guide → reference
0.902	0.111	1.56	cool, technology, computers → tech
0.902	0.222	2.80	cool, design, blogs → blog
0.900	0.172	1.21	cool, internet, free → web
0.900	0.123	1.38	webdesign, tips → web, design
0.900	0.062	5.40	web, development, web-design → html
0.900	0.124	3.07	design, css → webdev
0.900	0.074	2.13	web, osx → software
0.900	0.099	2.18	design, tutorials, css → development
0.900	0.025	6.75	software, mac, apple → osx
0.900	0.124	1.25	tutorial, resources → reference

The overlap between tags, resources and users is very sparse. To get a dense subset of the data we computed p-cores [11] for different levels. For $p = 100$ we get enough bookmarks for each resource to split the data into meaningful training and test sets (90%:10%). The test sets differ in the number of bookmarks each resource has assigned to simulate new resources that only have one to five user annotations. For this we removed all tags not belonging to the first n bookmarks, $n \in \{1 \ldots 5\}$.

Our final dataset consists of ∼10,000 resources, ∼10,000 users, and ∼3600 tags occurring in ∼3,200,000 tag assignments. We have five test sets containing 10% of the data. The 100-core ensures that each tag, each resource and each user appears at least 100 times in the tag assignments.

On this set, the only preprocessing of the tag assignments performed was the decapitalization of the tags. No stemming or other simplifications were applied. More sophisticated preprocessing might improve the results but would complicate the evaluation of the algorithms.

6.2.3 Results

In this Section we report results for our LDA-based algorithm and compare these with the numbers we get using association rules for the same task on the same dataset.

Association Rules For mining association rules, we have used RapidMiner [130]. For the 9000 resources in the training set we get almost 550 K association rules with a minimum support of 0.05 and a minimum confidence of 0.1, many of which are of course partially redundant. Table 6.3 gives the results for 5 bookmarks, at different confidence levels (Conf). Precision (Prec), recall (Rec), f-measure (FM) are measured at macro level, i.e., they are averaged over the individual measures for each resource. The maximum precision (Prec) of 0.648 for confidence ≥ 0.9 is lower than the 0.873 reported in [81], who operated on a bigger dataset (about $60K$ resources, split into $50K$ training and $10K$ testing). Maximum

Table 6.3: Results for tag recommendation using association rules with different minimum confidences and 5 known bookmarks

Conf	**Prec**	**Rec**	**FM**	**Avg TFIDF**	**Median TFIDF**
0.90	0.648	0.077	0.137	0.060	0.029
0.70	0.514	0.167	0.252	0.051	0.021
0.50	0.435	0.244	0.312	0.048	0.018
0.30	0.357	0.319	0.337	0.045	0.016
0.10	0.265	0.408	0.321	0.044	0.015

Table 6.4: Results for tag recommendation using association rules with minimum confidence 0.9 for 1–5 known bookmarks

#BM	**Prec**	**Rec**	**FM**	**Avg TFIDF**	**Median TFIDF**
1	0.741	0.041	0.077	0.054	0.030
2	0.691	0.056	0.104	0.057	0.030
3	0.682	0.066	0.120	0.059	0.029
4	0.663	0.072	0.130	0.060	0.029
5	0.648	0.077	0.137	0.060	0.029

f-measure is reached with association rules above the fairly low confidence of 0.3. The last two columns give the average and median TFIDF for correctly recommended tags. Both values lie in the same range as the corresponding values for the actual tags in the test set (0.054 and 0.018), which indicates that association rules tend to recommend rather general tags. In an attempt to recommend more specific tags, we have also experimented with a smaller support of 0.01. This however only increases recall at the cost of precision; the average and median specificity of recommended tags remains in the same range. For a smaller number of available bookmarks, precision goes up and recall goes down, and the f-measure slightly decreases. Average and median TFIDF remain essentially constant (see Table 6.4).

Latent Dirichlet Allocation The tag recommendation algorithm is implemented in Java. We used LingPipe[5], to perform the latent Dirichlet allocation with Gibbs sampling. The LDA algorithm takes three input parameters: the number of terms to represent a latent topic, the number of latent topics to represent a document, and the overall number of latent topics to be identified in the given corpus. After some experiments with varying the first two parameters we fixed them at a value of 100.

[5] `http://alias-i.com/lingpipe`

Table 6.5: Results for tag recommendation using LDA with 100 topics with different thresholds to recommend a tag for 5 known bookmarks

Thresh	**Prec**	**Rec**	**FM**	**Avg TFIDF**	**Median TFIDF**
0.01	0.717	0.174	0.281	0.169	0.079
0.005	0.609	0.245	0.349	0.140	0.057
0.001	0.370	0.439	0.401	0.096	0.031
0.0005	0.291	0.527	0.375	0.085	0.026
0.00001	0.168	0.669	0.269	0.071	0.022

Table 6.6: Results for tag recommendation using LDA with 100 topics and threshold 0.01 for 1–5 known bookmarks

#BM	Prec	Rec	FM	Avg TFIDF	Median TFIDF
1	0.680	0.069	0.126	0.233	0.128
2	0.717	0.112	0.193	0.199	0.097
3	0.712	0.139	0.233	0.186	0.089
4	0.711	0.160	0.261	0.174	0.084
5	0.717	0.174	0.281	0.169	0.079

Table 6.7: F-measure for different sized test set and different number of LDA topics (threshold 0.001)

#BM	# LDA topics			
	50	100	250	500
1	0.313	0.310	0.297	0.268
2	0.353	0.360	0.351	0.328
3	0.367	0.381	0.378	0.356
4	0.371	0.392	0.397	0.386
5	0.378	0.401	0.414	0.403

As described in Section 2.3.2 we can set a threshold for the probabilities up to which we recommended tags. Table 6.5 shows precision, recall, f-measure (FM), as well as average TFIDF and median TFIDF of the “correctly” recommended tags. Not surprisingly, precision decreases when lowering the threshold whereas recall increases. We get a maximum f-measure at 0.001 of 0.401

Table 6.6 gives detailed results for different numbers of known bookmarks using a threshold of 0.01 to recommend with high precision. Knowing more bookmarks in advance for a resource does not increase precision (2 bookmarks → 0.717; 5 bookmarks → 0.717) but increases recall significantly. The average TFIDF gives the expected value for the specificity of a tag whereas the median gives the typical specificity. Because the TFIDF values show a power law distribution, the average is of course larger than the median. Both values are significantly higher for tags recommended by LDA than by association rules, but also higher than the average and median TFIDF of the actual tags present in our tag set. As can be seen in Table 6.4 and Table 6.6 the TFIDF values are two to four times higher. Recommending resource specific tags with high TFIDF is particularly useful for search as pointed out in [19], fairly infrequent tags are usually used for topical and type annotations.

The results for varying the number of latent topics are shown in Table 6.7. The f-measure is shown for 50, 100, 250, and 500 latent topics. The number of bookmarks (#BM) indicates the number of users that have annotated a resource in the test set. The threshold for our recommendation was set to 0.001. As can be seen in the table, performance decreases with the LDA topic size for the 1 BM case. This effect is reversed when adding more bookmarks. A small number of topics typically leads to fairly general topics that are mixtures of more specific subtopics. Such general topics have a higher chance to be evoked by one of the few tags in one bookmark, leading to a higher recall. With more bookmarks, there are more tags, and it is more beneficial to separate the general topics into more specific topics. 100 LDA topics give the best average results.

Table 6.8 shows the actual tag distribution for a randomly selected resource (`http://www.connotea.org`), the top tags recommended by LDA with aggregated probabilities, and (all) the tags recommended by association rules based on five known bookmarks. The

Table 6.8: Actual tags with tag frequency and recommended tags with computed probablity for URL www.connotea.org

Real Tag	TF	TFIDF	LDA Tag	LDA Prob.	AR Tag	AR Conf.
science	0.0906	0.2281	***del.icio.us***	0.1001	***web***	0.912
bookmarks	0.0695	0.1721	***delicious***	0.0478	***reference***	0.760
tags	0.0546	0.1468	***tools***	0.0356	***tools***	0.664
reference	0.0521	0.0407	business	0.0223	***internet***	0.657
social	0.0509	0.1068	***language***	0.0204	cool	0.642
folksonomy	0.0447	0.1306	***bookmarks***	0.0166	tech	0.585
del.icio.us	0.0409	0.1166	***web***	0.0090	***software***	0.541
tools	0.0397	0.0271	***tool***	0.0090	toread	0.515
tagging	0.0360	0.1062	***science***	0.0085	technology	0.467
research	0.0347	0.0714	space	0.0065	interesting	0.417
delicious	0.0248	0.0722	dictionary	0.0064	design	0.398
bookmark	0.0236	0.0770	***bookmark***	0.0059	information	0.395
academic	0.0223	0.0780	english	0.0049	***search***	0.393
search	0.0223	0.0402	environment	0.0040	***blog***	0.391
web	0.0186	0.0084	***reference***	0.0039	–	–
bookmarking	0.0173	0.0717	astronomy	0.0037	–	–
tag	0.0161	0.0496	marketing	0.0033	–	–
socialsoftware	0.0149	0.0387	***tags***	0.0033	–	–
internet	0.0124	0.0124	cool	0.0032	–	–

tags available in the known bookmarks (first column)[6] and the correctly recommended tags (forth and sixth column) are marked in bold. As the actual tags indicate, Connotea is a tagging site focusing on scientists and scientific resources. The tags recommended by LDA come from five latent topics, comprising social systems, tagging, science, business, and language. These tags characterize Connotea quite well, and accordingly among the nine most likely recommended tags, there is only one rather general tag (business) that is not among the actual tags. In contrast, the tags recommended by association rules hardly characterize the site, but are rather non descriptive and general.

Tag Search To evaluate the effectiveness of our recommended tags for tag search we compared three result lists: The first is based on the test set with only 1 – 5 bookmarks per resource, the second uses the tags recommended by our algorithm. These two lists are compared with the list based on all original tags assigned to the test set. For the ranking of the results in each list, we implemented a simple baseline algorithm based on single keyword search. The resources are weighted according to the TFIDF score of the query tag. E.g. a search for the keyword "web" gives a list with resources annotated with the tag "web". The list is ranked according to tag frequency, i.e., how high is the number of "web"-tags compared to the overall number of tags assigned to a resource.

The test set without recommended tags is also ranked by TFIDF, whereas the test set with recommended tags is ranked by the probability assigned by LDA. To compare the three ranked lists, we need to first decide which of the baseline results are considered relevant. We report scores for taking the top 10 and the top 20 resources as relevant results. A well known measure for comparing rankings in information retrieval is Mean Average

[6] The first five bookmarks contain three more tags with rather low TF: webware, management, and social_software.

Table 6.9: Mean Average Precision (MAP) for tag search with and without extention of recommended tags

#BM	MAP for top 10	
	w/o Extend	w/ Extend
1	0.037	0.137
2	0.058	0.196
3	0.075	0.221
4	0.091	0.241
5	0.105	0.256

#BM	MAP for top 20	
	w/o Extend	w/ Extend
1	0.025	0.105
2	0.039	0.150
3	0.051	0.170
4	0.062	0.186
5	0.072	0.198

precision (MAP) [126], computed as follows:

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} \text{Precision}(R_{jk}) \tag{6.1}$$

with R_{jk} the set of ranked results from the top of the list down to item k in the list, where the set of relevant items is $\{i_1 \ldots i_{m_j}\}$. If no relevant document is retrieved, precision is taken to be 0.

Table 6.9 shows the MAP values based on the number of known bookmarks. When considering the top 10 TFIDF ranked results as relevant, extension of the resources with our recommended tags increases MAP by more than 300% for one known bookmark. When considering the top 20 results of the baseline algorithm as relevant, the MAP score for the LDA probabilities weighted ranked list increases by more then 400% in the one bookmark case.

6.3 Language Models and Topic Models for Personalized Tag Recommendation

In this section we present our approach to combine tag recommendation for users with tag recommendation for resources. We show that this combination helps to overcome the weaknesses of the individual approaches applied in isolation. On the one hand we take the user's interest and tagging preferences into account, and on the other hand we identify suitable tags for a particular resource. For both, user-centered and resource-centered, we investigate the use of two methods, latent Dirichlet allocation and language models for tag recommendation.

6.3.1 Approach

Tagging systems allow users to annotate resources with keywords. Tag recommendation aims at assistint the user with this task. As soon as the user decides to tag a new resource, the system suggests appropriate tags to alleviate the burden of coming up with new keywords and typing them for the user. The tag recommendation algorithm used in a system therefore also influences the tag distribution of resources since many users pick a recommended tag rather then conceiving new keywords. Hence, the recommendation algorithm is an important part of tagging systems.

The goal of personalized tag recommendation is to assist users bookmarking a new resource by reducing the cognitive load by suggesting tags for their bookmark $b(r, u)$. This can be based on other tag assignments to this resource and similar resources, or based on the user and similar users. To this end, we need to rank possible tags t, given a resource and a user. We rank based on a probabilistic approach. More formally, we estimate the probability $P(t \mid r, u)$ of a tag t given a resource r and a user u as follows:

$$\begin{aligned}
P(t \mid r, u) &= \frac{P(r, u \mid t)P(t)}{P(r, u)} && (6.2)\\
&\approx \frac{P(r \mid t)P(u \mid t)P(t)}{P(r, u)} && (6.3)\\
&= \frac{P(t \mid r)P(r)}{P(t)} \frac{P(t \mid u)P(u)}{P(t)} \frac{P(t)}{P(r, u)} && (6.4)\\
&= \frac{P(t \mid r)P(t \mid u)}{P(t)} \frac{P(r)P(u)}{P(r, u)} && (6.5)\\
&\propto \frac{P(t \mid r)P(t \mid u)}{P(t)} && (6.6)
\end{aligned}$$

Equation 6.2 applies Bayes' rule, Equation 6.3 splits $P(r, u|t)$ assuming conditional independence of r and u given t, Equation 6.4 again applies Bayes' rule to $P(r|t)$ and $P(u|t)$, Equation 6.5 simplifies, and Equation 6.6 discards the factors $P(r)$, $P(u)$, and $P(r, u)$, which are equal for all tags.

$P(t)$ can be estimated via the relative frequency of tag t in all bookmarks. For estimating $P(t|r)$ and $P(t|u)$ we investigate and combine two approaches. On the one hand, we use simple language models (Section 6.3.1), on the other hand, we use latent Dirichlet allocation (Section 6.3.1), in order to also recommend tags for new resources and users, which have only few bookmarks available.

The estimate in Equation 6.6 gives equal weight to $P(t|r)$ and $P(t|u)$. However, typically there are more tags available for a particular user u than for a resource r. Thus the estimate for $P(t|u)$ should be weighted more strongly than the estimate for $P(t|r)$. To this end, we smoothen $P(t|r)$ and $P(t|u)$ with the prior probability $P(t)$.

$$P'(t \mid r) \quad \propto \quad log_2(|r|+1)P(t \mid r) + log_2(|u|+1)P(t) \tag{6.7}$$

$$P'(t \mid u) \quad \propto \quad log_2(|u|+1)P(t \mid u) + log_2(|r|+1)P(t) \tag{6.8}$$

where $|r|$ is the number of tags available for a resource r, and $|u|$ is the number of tags available for a user u. When $|r|$ is smaller than $|u|$, $P(t|r)$ is smoothed more strongly, and thus influences $P(t|r,u)$ less than $P(t|u)$. Note that when $P(t|r)$ is zero, $P'(t|r)$ is proportional to $P(t)$. Consequently, the combined probability $P'(t|r) * P'(t|u)/P(t)$ is effectively proportional to $P'(t|u)$. Likewise, when a resource has no tags at all, $log_2(|r|+1) = 0$, and the combined probability is again proportional to $P'(t|u)$.

The combination above is reminiscent of the popular ”Product of Experts” approach, where our Experts are resources and users. We have also experimented with the popular mixture, which linearly interpolates $P(t \mid r)$ and $P(t \mid u)$. But this approach did not achieve competitive results.

Language Models The most straightforward approach to tag recommendation is to simply recommend the most frequent tags for each resource. More formally, the probability for a tag t given a resource r is estimated as:

$$P_{lm}(t \mid r) = \frac{c(t,r)}{\sum_{t_i \in r} c(t_i, r)} \tag{6.9}$$

where $c(t,r)$ is the count of tag t in resource r. The probability $P_{lm}(t \mid u)$ of a user u using tag t is determined in a similar way from all tags the user has assigned. This is the standard language model using maximum likelihood estimation, see Section 2.2.

Note that we do not need to smoothen the language models as usual, because we smoothen $P(t|r)$ and $P(t|u)$ with $P(t)$ via Equations 6.7 and 6.8.

Latent Dirichlet Allocation Especially for new resources and users with only few bookmarks, the simple language model does not suffice for tag recommendation, because the tag vocabulary of the already available bookmarks may differ from the preferred tag vocabulary of the user. Smoothing with the global tag probability only effectively switches off tags that are not available for a resource or user.

For resources, the resulting topics reflect a collaborative shared view of the resource, and the tags of the topics reflect a common vocabulary to describe the resource. Table 6.10 shows typical examples of resource topics (“Tech” and “Flickr”). As can be seen, the topics group typically co-occurring tags, which often will not be used by the same user. E.g., one user may prefer the tag ’photography’, another user may prefer ’photo’ or ’photos’.

For users, the resulting topics reflect the topical interests of a user, and the tags of topics reflect the individual tagging vocabulary of the user and similar users. Table 6.10 gives also examples of user topics (“Mac” and “DIY”). Note that latent topics are not necessarily disjoint. E.g. ’hardware’ occurs in the ’mac’ topic as well as in the ’do it yourself’ topic, but most certainly these two interpretations of ’hardware’ are rather disjoint.

By combining these two perspectives using Equation 6.6, the resource perspective serves as a selector of the topical content of the resources, while the user perspective takes into account the individual tagging practices of the user.

Table 6.10: Top tags composing the latent topics "tech news" and "flickr" based on resource profiles and "mac" and "do it yourself" based on user profiles

"Tech"-Topic		"Flickr"-Topic		"Mac"-Topic		"Diy"-Topic	
Tag	**Prob.**	**Tag**	**Prob.**	**Tag**	**Prob.**	**Tag**	**Prob.**
news	0.201	flickr	0.344	mac	0.320	diy	0.234
technology	0.182	photography	0.167	osx	0.215	make	0.099
tech	0.118	photos	0.117	apple	0.191	hardware	0.084
blog	0.082	photo	0.093	software	0.170	creativity	0.080
daily	0.070	tools	0.089	video	0.025	hacks	0.072
geek	0.067	web2.0	0.045	quicktime	0.013	electronics	0.070
blogs	0.029	visualization	0.016	macintosh	0.012	crafts	0.063
community	0.025	images	0.015	mail	0.012	science	0.046
internet	0.023	pictures	0.012	tv	0.009	mind	0.030
computers	0.021	api	0.010	ipod	0.006	theory	0.027
web	0.018	search	0.009	gmail	0.005	photography	0.023
forum	0.018	internet	0.007	hardware	0.004	engineering	0.019
computer	0.015	applications	0.005	algorithm	0.003	tutorials	0.017
software	0.013	sharing	0.004	boot	0.002	language	0.008

Combining LDA and LM As $P_{lm}(t \mid r)$ and $P_{lda}(t \mid r)$ both constitute (normalized) probability distributions, we can combine these two by straightforward linear interpolation (likewise for $P(t \mid u)$):

$$P(t \mid r) = \lambda \cdot P_{lm}(t \mid r) + (1 - \lambda) \cdot P_{lda}(t \mid r) \tag{6.10}$$

We have experimented with a broad range for λ, and achieved consistently good results for λ in the range of [0.2..0.8]. We report results for $\lambda \in \{0.0, 0.5, 1.0\}$, where $\lambda = 0$, and $\lambda = 1$ practically switch off the estimates based on language models and latent topics respectively. Combined with the smoothing in Equations 6.7 and 6.8, we effectively use a two level smoothing of the simple language model $P_{lm}(t \mid r)$: First by the more general $P_{lda}(t \mid r)$ and then by the marginal tag probability $P(t)$.

6.3.2 Evaluation

Evaluating personalized tag recommendation algorithms is not a trivial task. To get precise performance statistics a good way would be to compare two recommendation algorithms in a live tagging system, which allows for a direct user evaluation. Since this scenario is unfeasable or means interfering with a running system other approaches are prefered.

One popular way to evaluate is to take existing data from a tagging system and conduct tests on a hold-out set of tags, resources, or users ([79]). This approach has a promising characteristic: All tags which were used for a resource by a particular user are definitly known. The drawback is that these tags have been added by the user after being suggested by some automatic algorithm within the tagging system. This can bias the tag assignments towards the used tag recommendation algorithm, see [68]. Another disadvantage is that only a small set of correct, good tags are actually picked by the user making no distinction between totally unsuited tags and suitable recommended tags which were not picked by the user for whatever reason. Note that this leads to an underestimation of the actual tag recommendation quality.

To extenuate these disadvantages the test dataset has to be designed thoroughly. The strength of a recommendation algorithm can only be judged in comparison with other algorithms run on the same dataset. Thus we need to compare directly state-of-the-art algorithms with the proposed tag recommendation algorithm on the same dataset.

[67] observe that tag distributions for a resource tend to stabilize after around 100 bookmarks. This makes tag recommendation especially challenging for resources having a lot less bookmarks. This so-called cold start problem gives the most discriminative results for different algorithms.

Before we report our results, we have a detailed look into the used datasets, performance metrics, and the used baseline.

Datasets We performed experiments on two datasets. The first one is based on a crawl from Delicious. It consists of diverse urls tagged by Delicious users. The second dataset was provided in the context of a tag recommendation challenge held in conjunction with the ECML/PKDD conference 2009. It consists of data from the bookmarking system Bibsonomy, which not only includes tagged urls but also tagged Bibtex entries.

Delicious Dataset. We use a crawl from Delicious provided by [191]. The dataset consists of nearly 1 million users crawled between December 2007 and April 2008. The retrieval process resulted in about 132 million bookmarks or 420 million tag assignments that were posted between September 2003 and December 2007. Almost 7 million different tags are contained in the dataset and about 55 million different urls were annotated.

To do the computations in memory and in a reasonable time we were forced to use only a sample of the whole dataset. The huge amount of data and the fact that no spam filtering was done also results in a very sparse overlap between tags, resources and users. To get a dense subset of the sampled data we computed p-cores described by [11] for different levels.

For $p = 20$ we get enough bookmarks for each resource to split the data based on resources into meaningful training and test sets (90%:10%). The 20-core ensures that each tag, each resource and each user appears at least 20 times in the tag assignments. For the 10% resources in the test set, we only include the bookmarks for the first n users ($n \in 1, 3, 5, 7, 10, 20$) who annotated a resource into the training set. This results in a setting close to real life situations where users often annotate a resource previously annotated by only a few (n) other users. As soon as a resource is annotated by many users, tag recommendation can exploit the stabilized tag distribution ([67]) for resources and recommending good tags becomes less challenging.

The proposed setup allows to analyze how well different algorithms can generalize from relatively few tags available for a resource similuating the cold start problem in tagging environments.

Parameter settings were tested on 1/256 of the data. We have five test sets containing 10% of the resources with different numbers of "known" bookmarks. On this set, the only preprocessing of the tag assignments performed was the decapitalization of the tags. No stemming or other simplifications were applied. More sophisticated preprocessing improve the results but would complicate the evaluation of the algorithms and the comparison of different methods.

Bibsonomy Dataset. This dataset consists of the provided training and test data for the Discovery Challenge 2009 held in conjuction with ECML/PKDD 2009 ([55]). The training set consists of 253,615 tag assignments done by 1,185 individual users, 14,443 distinct URLs and 7,946 distinct BibTeX posts, and 13,276 distinct tags. This dataset was cleansed before by removing spammers and automatically added tags (like "imported", "public", "systemimported", etc.) and a post-core at level 2 was computed, that is, all users, tags, and resources which appeared in only one post were removed.

Three different tasks were provided aiming at different capabilities of the participating systems. Along with content-based and graph-based tag recommendation, one task dealt with online tag recommendation. Since our approach works without any additional content solely on the tags assigned by users to resources, the second task (graph-based tag recommendation) is predestined to test our algorithms on.

The test dataset for task 2 consists of 775 userId-contentId tuples extracted from the running Bibsonomy bookmarking system. For each userId-contentId tuple the participating systems had to recommend 5 tags. The actual tags assigned by the users are used as ground truth.

Evaluation Measures We use standard information retrieval evaluation metrics to report and compare the performance of the algorithms.

- *P@1 — precision at one:* Percentage of test cases where the first recommended tag was actually used by the user to annotate the resource. This is the same as success at one (S@1).
- *S@5 — success at five:* Percentage of test cases where at least one of the first five recommended tags was used by the user.
- *P@5 — precision at five:* Percentage of tags among the first five recommended tags that where actually used by the user. Averaged over all test cases.
- *F1@5 — f1 macro average at five:* The harmonic mean of averaged precision and recall for the first five recommended tags.
- *S@uAvg — success at user average:* Percentage of test cases where at least one of the recommended tags was used by the user. The number of recommended tags is the average number of tags per (other) bookmark for the user.
- *P@uAvg — precision at user average:* Percentage of tags among the top n recommended tags that where actually used by the user, where n is again the average number of tags per bookmark.
- *R@uAvg — recall at user average:* Percentage of user tags among the top n recommended tags, n as above.
- *MRR — mean reciprocal rank:* The average over all test cases of the multiplicative inverse of the rank of the first correct tag.

Baseline To get a good estimation of the performance of our tag recommendation algorithms we compare the results with the results from FolkRank by [84]. FolkRank (FR) is one of the state-of-the-art tag recommender algorithms. Its recommendations are very accurate but this comes with high computational costs. In contrast to our approach, FolkRank does not make use of latent topics but relies on a graph representation of the folksonomy.

The basic idea is to adapt PageRank by [141] to get scores for tags. A graph $G = (V, E)$ is constructed from the folksonomy $F = (U, R, T, Y)$, where the vertices are users, resources, and tags ($R \cup U \cup T$) and the weighted edges are co-occurrences of tags and users, tags and resources, and users and resources within tag assignments $(u, t, r) \in Y$.

The symmetric characteristic of the graph G would lead to scores biased towards "popular", i.e., highly connected entities within the graph when employing the adapted PageRank (ap). Thus, FolkRank uses a differential approach and computes the scores for each node based on the difference between a regular PageRank computation and a "personalized" PageRank, like, e.g., [73], using a preference vector.

Table 6.11: Results for one known bookmark and different algorithms on the Delicious dataset

Rec. based on		**P@1**	**S@5**	**P@5**	**F1@5**	**S@uAvg**	**P@uAvg**	**R@uAvg**	**MRR**
User	**Resource**								
FR	–	0.271	0.537	0.168	0.205	0.442	0.190	0.231	0.400
LDA	–	0.279	0.571	0.178	0.212	0.475	0.194	0.229	0.407
LM	–	0.284	0.584	0.187	0.225	0.482	0.204	0.246	0.424
LDA&LM	–	0.288	0.596	0.190	0.228	0.486	0.208	0.250	0.428
–	FR	0.488	0.768	0.281	0.337	0.657	0.294	0.376	0.601
–	LDA	0.496	0.815	0.310	0.370	0.675	0.328	0.416	0.635
–	LM	0.493	0.762	0.335	0.369	0.661	0.352	0.363	0.604
–	LDA&LM	0.560	0.826	0.334	0.397	0.718	0.358	0.454	0.678
LM	LM	0.547	0.813	0.319	0.382	0.726	0.353	0.441	0.667
LDA	LDA	0.561	0.847	0.336	0.404	0.738	0.370	0.467	0.689
LDA	LM	0.532	0.812	0.313	0.375	0.722	0.340	0.425	0.653
LM	LDA	0.566	0.859	0.343	0.416	0.759	0.386	0.488	0.703
FolkRank		0.570	0.840	0.325	0.393	0.734	0.354	0.452	0.689
LDA&LM	LDA&LM	0.610	0.890	0.372	0.448	0.795	0.415	0.529	0.733

For tag recommendation this preference vector is highly biased towards two entries: the user and the resource for whom the recommendation is computed, see [93]. To compare FolkRank with LDA and LM employed only on resources or only on users, we only boost one entry in the preference vector — the resource or the user in question. This gives either resource-centered or user-centered FolkRank results.

The FolkRank scores are computed iteratively and finally combined:

$$R_{i+1}^{ap} = c(\alpha R_i^{ap} + \beta A R_i^{ap}) \tag{6.11}$$

$$R_{i+1}^{pref} = c(\alpha R_i^{pref} + \beta A R_i^{pref} + \gamma P) \tag{6.12}$$

$$R = R^{pref} - R^{ap} \tag{6.13}$$

where α, β, γ are constants and c is a normalization factor. A is a row-stochastic version of the adjacency matrix of G. Since the first summand does not influence the result, it is unnecessary to compute it. Thus, setting $\alpha = 0$ speeds-up the convergence.

6.3.3 Results

We have systematically applied the described approaches for estimating $P(t|r)$, $P(t|u)$, and $P(t|u, r)$, and compared them to the corresponding results using FolkRank. We first report the achieved results for the Delicious dataset and then for the Bibsonomy dataset.

Results for Delicious Dataset Table 6.11 gives a complete overview of the results for tag recommendation when there is only one bookmark available for the resource. In this scenario a user introduces a new resource into the system and adds some initial tags.

The first four rows give the results for taking only the user perspective into account, i.e., tags are predicted based on $P(t|u)$ only (or for FolkRank the preference vector is only biased towards the user). We see that generally the probabilistic approach introduced

Table 6.12: F-macro average for different number of known bookmarks on the Delicious dataset

Known	F-Macro Average	
Bookmarks	FolkRank	LDA+LM
1	0.393	0.448
3	0.447	0.476
5	0.462	0.477
10	0.475	0.491

in this section outperforms FolkRank (FR) w.r.t. all measures. The simple (smoothed) language model approach (LM) slightly outperforms LDA. The linear interpolation of LM with LDA achieves a very slight improvement over LM and LDA alone. It is also clear that the user perspective in isolation performs worse than the resource perspective (next four rows). This is to be expected. The mixture of topics that a user is interested in is typically much more diverse than the mixture of topics a particular resource is about. Thus, no matter how the tag probabilities are estimated, just recommending the most likely tags for a user, disregarding the resource, will often go astray.

The general trends for tag recommendation based on resources only (second four rows) are slightly different. FR and LM are rather clearly outperformed on all measures, but among them, for some measures FR is better than LM and vice versa. LDA comes on a clear second place, and a very clear winner is is again the linear interpolation of LM and LDA.

It is interesting that LDA outperforms LM on resources, while LM outperforms LDA on users. The strength of LDA is to generalize from the tagging practices of individual users who have assigned tags to a particular resource (such as “photography”), in order to also include semantically related tags (such as ”photo”). This strength turns out to be a slight weakness for the user perspective, possibly because users tend to stick to a particular vocabulary, and thus the generalization by LDA does not help.

The next four rows inspect the performance of the individual approaches to estimate the probability of a tag when combined for personalized tag recommendation. The most simple (and scalable) approach by just combining the smoothed language models already achieves significant improvements[7] compared to tag recommendation based on LMs for the resource only. Combining only LDA for the user and resource yields further improvement, and the best combination of individual models is achieved by using LM for the user perspective and LDA for the resources. This is consistent with the results for user-based (LM best) and resource-based recommendation (LDA best).

Finally, the last two rows compare full FolkRank with a complete combination of user-based recommendations using an interpolation of LDA and LM with resource-based recommendation. This full combination outperforms FolkRank signifantly on all measures. All relative improvements are in the range of 7 % to 17 % with 11 % average.

Table 6.12 shows the F-macro average comparing the performance having differing prior knowledge about an item.

Figure 6.1 compares mean reciprocal rank (MRR) of the main approaches when varying the number of available bookmarks between 1 and 20. FR stands for FolkRank, LM for a combination of language models for the user and the resource, LDA and LM for the full combination of language models and LDA on users and resources. The full combination clearly outperforms the other two approaches, but notably the scalable combination of

[7] all improvements are significant well beyond a confidence of 0.99 based on a 2-tail paired t-test.

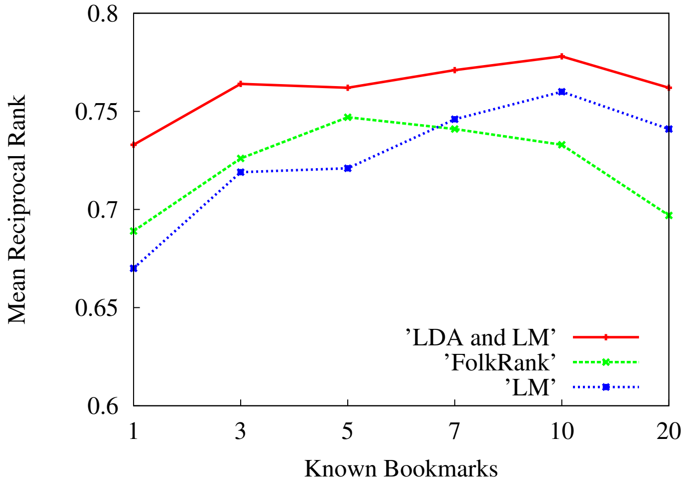


Figure 6.1: MRR for different numbers of known bookmarks on the Delicious dataset

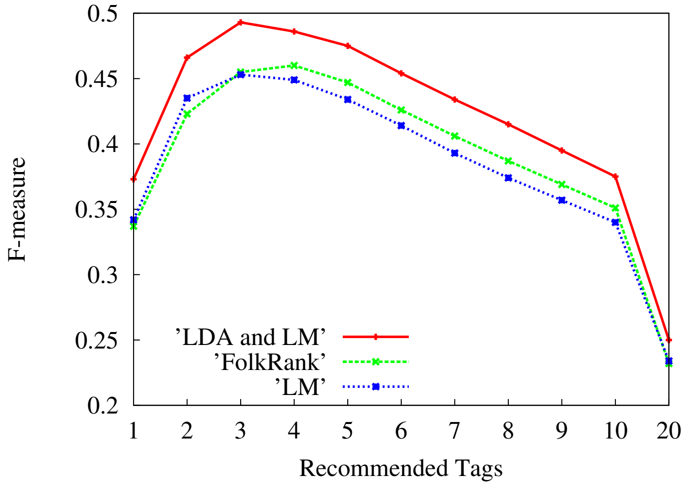


Figure 6.2: Macro f-measure for different numbers of recommended tags on Delicious data

simple language models outperforms FolkRank for more than seven bookmarks. The reason why MRR degrades with all approaches at least for 20 bookmarks is due to the experimental setup. When using 20 bookmarks, much fewer test data are available in the post-core at 20, and the few remaining test data may be the most difficult ones.

Finally, Figure 6.2 shows the progression of F-Measure depending on the number of recommended tags for resources with three known bookmarks. For all approaches, recommending three tags appears to provide the best balance between recall and precision. This also reflects the tagging behaviour of users who on average assigned 4.3 tags to one resource in our dataset. Again the approach presented in this section clearly outperforms FolkRank and smoothed language models, with the latter two being more or less on par.

To get an impression of the actual tags recommended, Table 6.13 gives a randomly picked example of tags recommended by FolkRank and by the approach introduced in this section. The correctly predicted tags are in bold. We see that our approach correctly predicts 4 tags in top 6, and 6 correct predictions in the top 20,whereas FolkRank predicts only 4 in top 20. But of course a single example can only provide anecdotal evidence. For

Table 6.13: Recommended tags for user 800 and resource "www.xfront.com/microformats/" from the Delicious dataset

Original tags from user
webdev
programming
reference
web2.0
webdesign
xhtml
microformats
tutorial

FolkRank		**LDA&LM**	
Tags	Score	Tags	Score
microformats	0.0138	**microformats**	50.6
howto	0.0078	howto	15.1
standards	0.0070	**tutorial**	12.8
tutorial	0.0068	standards	11.5
collection	0.0066	**programming**	9.6
information	0.0064	**reference**	8.4
resources	0.0061	semantic	7.2
tutorials	0.0060	development	5.3
webdev	0.0024	software	4.0
tool	0.0021	web	3.4
development	0.0020	xml	3.2
programming	0.0018	**webdesign**	3.0
web	0.0013	code	2.7
html	0.0011	tool	2.4
code	0.0009	**webdev**	2.3
software	0.0007	information	2.3
javascript	0.0006	css	2.0
python	0.0005	design	2.0
snippets	0.0004	tips	2.0
optimization	0.0004	tutorials	1.6

tag recommendation both tag sets make intuitively sense and the underestimation of the hold-out strategy can be observed.

Results for Bibsonomy Dataset An overview of the performance of the algorithms on the Bibsonomy dataset is presented in Table 6.14. Since the official task required five tags to be recommended for each user–resource pair in the test set, we only report precision@5, recall@5, and f1@5. The first four lines in the table show the results for only using the user profiles to recommend tags. As with the Delicious dataset, the results for using only the resource profiles are nearly three times as good. Using language models only on the resource information is already enough to beat FolkRank. Adding LDA and the knowledge about the tagging behaviour of a user in the past just slightly improves the results (from 0.301 to 0.308). This is due to the characteristics of this dataset. The resources are already tagged by many users and the tag distribution for a single resource has already stabilized. We are not dealing with a cold start problem in this dataset which lowers the probability that LDA can find very relevant tags that have not been used by users for a resource so far.

Another dataset dependent parameter is the number of latent topics. Table 6.15 depicts the f1-measure for using between 100 and $5,000$ latent topics for the corpus. A maximum of the performance is reached by $1,000$ topics. The table also indicates, that recommending 4 tags gives better recall and precision values then recommending 5 tags. The reason for this is the average number of tags a user assigns to a resource in the Bibsonomy system (3.96 tags/resource). By adapting our algorithm to the average number of tags a user assigns, i.e. recommending not 5 tags for all users but less if the average number of tags a user assigned to a resource was less than 5, we can even improve the F1@5 score. Of course recall will drop a little, but precision will be significantly higher.

To see the impact of the user profile information and the resource profile information,

Table 6.14: Results on Bibsonomy dataset for five recommende tags

Rec. based on		**Prec@5**	**Rec@5**	**F1@5**
User	**Resource**			
FR	–	0.079	0.124	0.096
LM	–	0.083	0.125	0.100
LDA	–	0.084	0.130	0.102
LDA&LM	–	0.084	0.129	0.102
–	LDA	0.209	0.318	0.252
–	FR	0.238	0.365	0.288
–	LM	0.258	0.361	0.301
–	LDA&LM	0.253	0.384	0.305
LM	LM	0.218	0.334	0.264
LDA	LDA	0.218	0.336	0.265
LM	LDA	0.215	0.339	0.263
LDA	LM	0.230	0.350	0.270
FolkRank		0.241	0.376	0.294
LDA&LM	LDA&LM	0.252	0.394	0.308

Table 6.15: F-measure for different number of recommended tags and different number of LDA topics using LM&LDA on resources and LM&LDA on users for the Bibsonomy dataset

No. Tags	**# LDA topics**					
	100	200	500	1,000	3,000	5,000
1	0.219	0.208	0.230	0.221	0.222	0.214
2	0.286	0.280	0.292	0.295	0.288	0.277
3	0.301	0.309	0.311	0.313	0.301	0.306
4	0.308	0.313	0.311	0.318	0.307	0.310
5	0.303	0.311	0.305	0.315	0.303	0.306

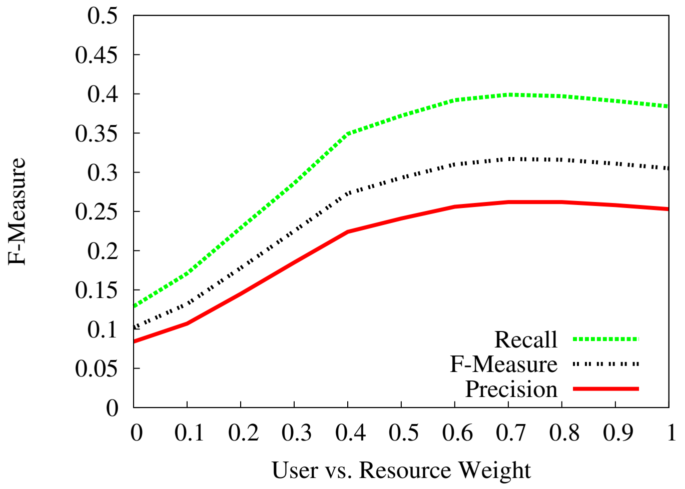


Figure 6.3: Precision, recall, and f-measure for different weights on user and resource information on the Bibsonomy dataset

we plotted the f1 values for different weights α in Figure 6.3. The weighting is done similar to Equation 6.10:

$$P(t \mid r, u) = \alpha \cdot P_{lm\&lda}(t \mid r) + (1 - \alpha) \cdot P_{lm\&lda}(t \mid u) \tag{6.14}$$

where α defines the weight for combining the resource with the user information. A maximum for recall as well as for precision is found for $\alpha = 0.7$. But in general the resource information is much more valuable in this setting than the user profiles.

In this approach, we are only using the tag assignment information and no meta-information or content in any way. This makes our approach universal with respect to the underlying tagging system. On the other hand, different systems, like for example the bookmarking system Bibsonomy, offer more information that could be used. This information can be very valuable when recommending tags. The best tag recommendation systems at the discovery challenge [55] exploited this additional information, such as content of the tagged resource or meta information like a resource description. A recent study by [120] analyzed the use of resource titles to tag these resources. It shows the benefits that can be gained by recommender algorithms taking these tagging system dependent information into account.

6.4 Related Work

In recent years interest in tag recommendation was sparked within the research community. The growing importance of tagging systems led to the development of sophisticated tag recommendation algorithms. The various approaches applied for the Data Discovery Challenge 2009 [55] represent a good overview.

Collaborative Filtering A popular approach to tag recommendation has been collaborative filtering [79], taking into account similarities between users, resources, and tags. Mishne [133] introduces an approach to recommend tags for weblogs, based on similar weblogs tagged by the same user. Chirita et al. [36] realize this idea for the personal desktop, recommending tags for web resources by retrieving and ranking tags from similar documents on the desktop.

Jäschke et al. [93] compare two variants of collaborative filtering and FolkRank [84], a graph based algorithm for recommendations in folksonomies. For collaborative filtering, once the similarity between users on tags, and once the similarity between users on resources is used for recommendation. FolkRank uses random walk techniques on the user-resource-tag (URT) graph based on the idea that popular users, resources, and tags can reinforce each other. These algorithms take co-occurrence of tags into account only indirectly, via the URT graph. Our evaluation shows that our approach achieves significantly better accuracy than FolkRank, and even the simple and scalable combination of smoothed language models achieves competitive accuracy.

Xu et al. [202] describe a way to recommend a few descriptive tags to users by rewarding co-occurring tags that have been assigned by the same user, penalizing co-occurring tags that have been assigned by different users, and boosting tags with high descriptiveness. An interactive approach in the context of a photo tagging site based on co-occurrence is presented in [62]. After the user enters a tag for a new resource, the algorithm recommends tags based on co-occurrence of tags for resources which the user or others used together in the past. After each tag the user assigns or selects, the set is narrowed down to make the tags more specific. Sigurbjörnsson and van Zwol [169] also look at co-occurrence of tags to recommend tags based on a user defined set of tags. The co-occurring tags are then ranked and promoted based on e.g. descriptiveness.

Heymann et al. [81] employ association rule mining on the tag sets of resources for *collective* tag recommendation. The mined association rules have the form $T_1 \rightarrow T_2$, where T_1 and T_2 are tag sets. On this basis tags in T_2 are recommended, when all tags in T_1 are available for the resource, and the confidence for the association rules is above a threshold. In Section 6.2.3 we have shown that tag recommendation based on LDA achieves significantly better accuracy than this approach, and recommends more specific tags, which are more useful for tag-based search.

Wetzker et al. [192] introduce an approach for personalized tag recommendation based on tensor calculus. Their approach is similar to the approach based on language models presented in Section 6.3, but differs with respect to normalization of tag weights and the way, the resource perspective is taken into account. By using a more principled probabilistic approach for combining the resource perspective with the user perspective, our approach can benefit more readily from better estimates of tag probabilities, based, e.g., on latent Dirichlet allocation.

Clustering A general problem of tagging systems is their sparsity. This has led to a number of approaches using clustering to map the sparse tagging space to fewer dimensions.

Symeonidis et al. [173] employ dimensionality reduction to personalized tag recommendation. Whereas [93] operate on the URT graph directly, [173] use generalized techniques of SVD (Singular Value Decomposition) for n-dimensional tensors. The 3-dimensional tensor corresponding to the URT graph is unfolded into 3 matrices, which are reduced by means of SVD individually, and combined again to arrive at a more dense URT tensor approximating the original graph. The algorithm then suggests tags to users, if their weight is above some threshold. Rendle and Schmidt-Thieme [159] introduce two more efficient variants of this approach using canonical decomposition and pairwise interaction tensor factorization. These tensor based techniques can be readily compared to our approach. The user-tag and resource-tag perspectives combined in this chapter correspond to 2 of the 3 matrices, and the LDA can be seen as an alternative dimensionality reduction technique. Indeed, on the Bibsonomy dataset (see Section 6.3.2), both approaches achieve similar accuracy.

When content of resources is available, tag recommendation can also be approached as a classification problem, predicting tags from content. A recent approach in this direction is presented in [171]. They cluster the document-term-tag matrix after an approximate dimensionality reduction, and obtain a ranked membership of tags to clusters. Tags for new resources are recommended by classifying the resources into clusters, and ranking the cluster tags accordingly.

In [167], Shepitsen et al. propose a *resource* recommendation system based on hierarchical clustering of the tag space. The recommended resources are identified using user profiles and tag clusters to personalize the recommendation results. Using LDA topic models to recommend resources rather than tags is subject for future work.

Tag Search Tags have been proven to be very useful for search: in case of image search where content based features are very difficult to extract [46], in case of enterprise search where not enough link information is available [54], or in case of web search to optimize results [10]. A large scale evaluation of Delicious regarding search is presented in [80]. They found that 50% of the pages annotated by a particular tag contain the tag within the page's content. Bischoff et al. [19] provide an in-depth analysis of a number of tagging systems with respect to to their usefulness for search. They observe that descriptive tags such as topic or type tags are much more frequent than personal tags such as ”to read”, especially in the mid and low tag frequency range, and that these tags are indeed used in search. Berendt and Hanser [14] argue that tags can be considered content and not just metadata which makes them valuable in a content based retrieval scenario as well.

Recently a number of papers deal with imroving search in tagging systems. Krestel and Chen [104] propose a method to measure the quality of tags with respect to the annotated resource to identify high quality tags that describe a resource better than others. Hotho et al. [84] propose exploiting co-ocurrence of users, resources, and tags for searching and ranking within tagging systems. “FolkRank” is using a graph model to represent the *folksonomy* and can be used to rank classical keyword search results. In [12], Begelman et al. present a tag clustering algorithm to improve search. The setting is similar to ours: Related tags are identified that can be used for extending existing resource annotations, query expansion or result clustering. The clustering is based on simple co-occurrence counts. Unfortunately, the paper does not contain a sound evaluation of the results. Schenkel et al. [165] propose to improve search in tagging systems by expanding a user query with semantically similar tags and rank the result additionally based on a social component, which means that tagging information of friends of a user in the network is taken into account when a user submits a query.

LDA for Tag Recommendation Latent Dirichlet allocation, a variant of clustering in particular suitable for bag of words data, has recently gained some attention for tag recommendation. An approach to *collective* tag recommendation using LDA is introduced in [168]. They employ LDA for eliciting topics from the words in documents (blogposts) and from the associated tags, where words and tags form disjoint vocabularies. On this basis they recommend new tags for new documents using their content only. Krestel et al. on the other hand use LDA to infer topics from the available tags of resources and then recommend additional tags from these latent topics. In this chapter we extend these approaches for *personalized* tag recommendation by also taking the personal tagging practices of users into account. Moreover, we show that using a mixture of language models and latent topic models significantly improves the accuracy of tag recommendation.

Bundschus et al. [29] introduce a combination of LDA based on the content and tags of resources and the users having bookmarked a resource. The underlying generative process elicits *user specific* latent topics from the resource content and seperately from the tags of the resource. The content-based topics and tag-based topics are in a one-to-one correspondence by the user-id. On this basis personalized tag recommendation is realized by first eliciting user specific topics from the resource content, and then using the corresponding tag-based topics for suggesting tags. Our approach does not require content, which may not be available, e.g., for multimedia data, but works exlusively on the tags.

Harvey et al. [77] introduce a similar approach to personalized tag recommendation as proposed in this chapter on the basis of LDA. Rather than decomposing the joint probability of a *tag* given the tag assignments for a resource and a user via an application of Bayes' rule (see Equation 6.6), they decompose the joint probability of *latent topics* given the tag assignments. On this basis, they introduce an extended Gibbs sampler which draws topics simultaneously from the user and the resource. This fully generative approach, however, requires some initial tags from the user to a given resource, in order to recommend additional tags. In contrast, our approach can also handle the arguably more realistic setting of suggesting tags for a new bookmark without any initial tags from the user.

Our Contributions. In the first part of this chapter we have investigated the use of latent Dirichlet allocation for collective tag recommendation. Compared to association rules, LDA achieves better accuracy, and in particular recommends more specific tags, which are more useful for search. In general, our LDA-based approach is able to elicit a shared topical structure from the collaborative tagging effort of multiple users, whereas association rules are more focused on simple terminology expansion. However, both approaches succeed to some degree in overcoming the idiosyncracies of individual tagging practices.

In the second part of this chapter we have explored user-centered and resource-centered approaches for personalized tag recommendation. We compared and employed a language modeling approach and an approach based on latent Dirichlet allocation. We furthermore thoroughly investigated the use of language models and LDA for tag recommendation showing that simple language models built from users and resources yield competitive performance while consuming only a fraction of the computational costs compared to more sophisticated algorithms. We showed that the combination of both methods (LDA and LM) tailored to users and resources outperforms state-of-the-art tag recommendation algorithms with respect to a broad variety of performance metrics.

CHAPTER
SEVEN

SUMMARY AND OUTLOOK

7.1 Summary

In this thesis we investigated and discussed the use of language models and topic models as a way to represent natural language Web data. We covered a broad spectrum of typical Web applications to show the effectiveness and benefits of the two representations.

Topic models are generative models that introduce an additional layer of information between single documents and single words. This layer contains the latent topics in a collection and consists of clustered terms. For many applications, this clustering effort pays off in terms of more structured data, reduced data space, and easier to comprehend data models. The generative character of topic models also allows to analyze corpora and use information about the detected latent topics. Language models on the other side are easy to understand and can be implemented very efficiently. Sparsity and idiosyncrasy hinder on the one hand the discovery of new knowledge but on the other had are useful for personalization tasks.

We chose four characteristic Web applications: filtering, classification, ranking, and recommendation to evaluate the use of topic models and language models.

For *filtering* we analyzed the possibility to distinguish and predict important and unimportant news. The language model approach is based on training of a SVM classifier using a bag-of-words representation. For the topic model approach we used the detected latent topics as features for the SVM classifier. This not only improved the prediction accuracy but also helped to analyze and explain the results indicating which topics in a news article makes it more likely to be important.

Within the large area of *classification* we looked into sentiment classification and how topic models can be incorporated to build context-dependent sentiment lexica. We used latent topics to represent product review data resulting in a fuzzy categorization of the review texts. The context is defined by the topics and an adapted version of pointwise mutual information was used to identify term polarity and sentiment scores. We showed that topic models can be used successfully to represent topical context and enable context-dependent sentiment classification.

Ranking, which plays a crucial part in the Web, was highlighted in combination with diversification. We showed the use of language models and topic models for Web search result diversification and evaluated the topical coverage of the top-k results. Topic models exhibited a slightly better performance than using language models as the underlying document representation and additionally helped to explain the reranking process while accounting for less reranking compared to language models. For product review ranking diversification, topic models were employed to ensure coverage of more product features within the top-k reviews. The latent topics cover different aspects of a product and make the knowledge about them a valuable asset.

Finally we investigated the use of topic models and language models to improve tag *recommendation* within folksonomies. Latent topics proofed to be very successful to deal with the cold start problem allowing to recommend tags to a user that were neither used before for the resource nor by the user. Language models outperformed topic models when it comes to personalized tag recommendation. The best results were achieved by combining both approaches making use of the user's profile as well as the latent topics present in the folksonomy.

For all four Web applications the use of topic models added valuable information and provided a fuzzy structure of the data. While the performance gains were not always significant, latent topics facilitated the comprehension of the results and gave an overview of the utilized data. To build a topic model for a dataset is more costly than using a language model representation directly which leads to a trade-off between cost and gain that needs to be done depending on the task and application.

7.2 Outlook

Topic models are a rather new method for dimensionality reduction. As a generative model they also can help to explain the generation of text and give insights into the characteristics of a corpus and word relations. Further they can help to analyze documents and structure document collections.

Since the Web, especially user-generated Web 2.0 content and semantic Web content, is growing in terms of textual content, topic models are an active research field for Web data. There are still many open research questions: for example, the right number of latent topics in a corpus [6, 71] or how to evaluate topic models [186, 31].

There are interesting application scenarios[20] to be explored like using topic models for visual data[187], micro-blogging [157], visualization [135], or summarization [87, 74]. Besides extending the method to various application domains, new topic models and improved parameter estimation methods are being developed.

Language models on the other hand are still the best choice for many machine learning or classification task where the result is more important than the analysis of the data. The very good results we achieved in personalized tag recommendation by combining topic models and language models raises the question whether other application domains would also profit from a combination of both.

Filtering The biggest open issues concerns time. Our current approach has no time dimension. Incorporating the temporal aspects of news articles is decisive to improve accuracy further. Not only because “nothing is older than yesterday’s news”, but also because of the interdependence of news stories. A news story might not be important only based on its content but also because there was another news story related to it. For future work we try to involve these temporal aspects of news. Extending the LDA implementation to consider a time dimension might be necessary to achieve this. We also try to incorporate more knowledge from the domain of journalism where research on importance factors of news articles has been carried out. A detailed analysis of mentioned numbers or dates with articles might also improve accuracy. Clearly, adding more training data increases prediction accuracy, we thus try to build up a larger corpus. We also try to improve the fulltext extraction of the articles from the websites to get less noisy data. There is significant scope in improving accuracy, and we continue to explore additional features which will make this possible.

Classification For classification, we are interested in testing alternative methods for discriminative analysis such as L_1 regularization and other feature selection techniques. In addition, we want to explore methods to find the optimal number of latent topics for a corpus a priori. Introducing special rules for sentences containing negations might lead to further improvements. Furthermore, it might also be interesting to distinguish between general sentiment terms and sentiment terms relying on context. Finally, we believe that a promising area for future research is the combination of our method with techniques for sentiment lexicon generation based on term co-occurrences, grammatical analysis, seed sets of manually assessed sentiment terms, or glossary descriptions. We consider our method as orthogonal to these approaches; alternative methods can be complemented but do not become obsolete.

Ranking We want to apply result diversification in the context of summarization of search results as well as of events in blogs and newspaper articles. Moreover, we want to experiment with using cross-entropy and Kullback-Leibler divergence directly for reranking search results such that the top-k results provide a representative overview on the complete result. Finally, we also want to develop approaches to diversification and evaluation, which better focus on the topical content of documents.

Regarding product review ranking, we will investigate the possibility of personalizing the review rankings by taking personal preferences of users into account. For example, a user might be more interested in the battery life of a product than the screen size. Another interesting direction is analyzing and categorizing product reviews on a large scale to identify different types of reviews. As trust is a factor that directly affects the user confidence, we will investigate the possibility of personalizing the review rankings by taking the trust between users and authors of reviews into account.

Recommendation Combining different approaches could be interesting to improve recommendation further. We plan to investigate whether it is beneficial to combine association rules and LDA, for example. As we showed in Section 6.2.3 the tags that are recommended by both algorithms differ significantly from each other. Our hypothesis is that accuracy can be improved by combining the more general tags recommended by association rules with the more specific tags recommended by LDA. Along similar lines, we also plan to investigate combining language models derived from the actual tags annotated to a resource with the latent topic models.

The main contribution of latent topic models is to reduce sparsity of the tag space. This gives rise to several interesting lines of research we will investigate: Mapping resources to their latent topics may result in more robust resource recommendation. Eliciting latent topics from the tagging practices of individual users and combining them with the latent topics for resources is a promising direction for personalized tag recommendation. Finally, we will experiment with using the probability of tags derived from topic models for visualizing tag recommendations in the form of tag clouds.

Regarding datasets, we also want to experiment with datasets from different domains, to check whether photo, video, or music tagging sites show different system behavior influencing our algorithms.

Further, we want to investigate the use of LDA and language models for item or user/community recommendation in the context of tagging systems. Especially for item recommendation, the extension of our approach to incorporate content information might be beneficial. Even for non-textual resources like videos or audio, additional metadata could be exploited. It would also be interesting to see whether the behavior of the current algorithms changes when applied to a photo or video tagging system instead of bookmarking systems. One question in this context would be whether users tag videos differently then web pages and whether LDA and LM can be employed in the same manner. Also the different characteristics of photo and bookmarking sites regarding tagging behavior could influence the algorithms. Finally, we plan to investigate how additional contextual knowledge such as time, location, and current task can be used to further personalize tag recommendation. A starting point to this end could be to have a multi-lingual aware, personalized tagging system dealing with identification of users' native languages and possibly automatic translation of tags.

BIBLIOGRAPHY

[1] Silvana Aciar, Debbie Zhang, Simeon Simoff, and John Debenham. Informed recommender: Basing recommendations on consumer product reviews. *Intelligent Systems, IEEE*, 22(3):39–47, May–June 2007.

[2] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14, New York, NY, USA, 2009. ACM.

[3] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993*, pages 207–216. ACM Press, 1993.

[4] James Allan, Jay Aslam, Nicholas Belkin, Chris Buckley, Jamie Callan, Bruce Croft, Sue Dumais, Norbert Fuhr, Donna Harman, David J. Harper, Djoerd Hiemstra, Thomas Hofmann, Eduard Hovy, Wessel Kraaij, John Lafferty, Victor Lavrenko, David Lewis, Liz Liddy, R. Manmatha, Andrew McCallum, Jay Ponte, John Prager, Dragomir Radev, Philip Resnik, Stephen Robertson, Roni Rosenfeld, Salim Roukos, Mark Sanderson, Rich Schwartz, Amit Singhal, Alan Smeaton, Howard Turtle, Ellen Voorhees, Ralph Weischedel, Jinxi Xu, and ChengXiang Zhai. Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, university of massachusetts amherst, september 2002. *SIGIR Forum*, 37:31–47, April 2003. ACM. New York, NY, USA.

[5] Alina Andreevskaia and Sabine Bergler. When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 290–298. The Association for Computer Linguistics, 2008.

[6] R. Arun, V. Suresh, C. E. Veni Madhavan, and M. N. Narasimha Murthy. On finding the natural number of topics with latent dirichlet allocation: Some observations. In Mohammed Javeed Zaki, Jeffrey Xu Yu, B. Ravindran, and Vikram Pudi, editors, *Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part I*, volume 6118 of *Lecture Notes in Computer Science*, pages 391–402. Springer, 2010.

[7] 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.

[8] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association, 2010.

[9] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[10] Shenghua Bao, Gui-Rong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy, editors, *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 501–510, New York, NY, USA, 2007. ACM.

[11] Vladimir Batagelj and Matjaz Zaversnik. Generalized cores. *CoRR*, cs.DS/0202039, 2002.

[12] Grigory Begelman, Philipp Keller, and Frank Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Proceedings of the WWW 2006 Workshop on Collaborative Web Tagging*, May 2006.

[13] Philip Beineke, Trevor Hastie, Christopher Manning, and Shivakumar Vaithyanathan. An exploration of sentiment summarization. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, Stanford, US, 2003.

[14] Bettina Berendt and Christoph Hanser. Tags are not metadata, but just more content - to some people. In *Proceedings of the International Conference on Weblogs and Social Media*, 2007.

[15] Indrajit Bhattacharya and Lise Getoor. A latent dirichlet model for unsupervised entity resolution. In *SIAM Conference on Data Mining (SDM)*, pages 47–58, April 2006.

[16] Bodo Billerbeck, Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, and Ralf Krestel. Exploiting click-through data for entity retrieval. In *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 168–169. ACM, July 19–23 2010.

[17] Bodo Billerbeck, Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, and Ralf Krestel. Ranking Entities Using Web Search Query Logs. In *Research and Advanced Technology for Digital Libraries, 14th European Conference, ECDL 2010, Glasgow, UK, September 6-10, 2010. Proceedings*, volume 6273 of *Lecture Notes in Computer Science*, pages 273–281, Berlin, Heidelberg, September 6–10 2010. Springer.

[18] István Bíró, Dávid Siklósi, Jácint Szabó, and András A. Benczúr. Linked latent dirichlet allocation in web spam filtering. In *AIRWeb '09: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, pages 37–40, New York, NY, USA, 2009. ACM.

[19] Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. Can all tags be used for search? In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 193–202, New York, NY, USA, 2008. ACM.

[20] David Blei, Jordan Boyd-Graber, Jonathan Chang, Katherine Heller, and Hanna Wallach, editors. *NIPS 2009 Workshop on Applications for Topic Models: Text and Beyond*, Whistler, Canada, December 2009.

[21] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*. MIT Press, 2004.

[22] David M. Blei and John D. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, 2005.

[23] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 113–120, New York, NY, USA, 2006. ACM.

[24] David M. Blei and Jon D. McAuliffe. Supervised topic models. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*. MIT Press, 2008.

[25] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

[26] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, Boomboxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447. Association for Computational Linguistics, June 2007.

[27] Dirk Bollen and Harry Halpin. An experimental analysis of suggestions in collaborative tagging. In *2009 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2009, Milan, Italy, 15-18 September 2009, Main Conference Proceedings*, pages 108–115. IEEE, 2009.

[28] Juergen Bross and Heiko Ehrig. Generating a context-aware sentiment lexicon for aspect-based product review mining. In *WI-IAT '10: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Washington, DC, USA, August 31–September 3 2010. IEEE Computer Society.

[29] Markus Bundschus, Shipeng Yu, Volker Tresp, Achim Rettinger, Mathaeus Dejori, and Hans-Peter Kriegel. Hierarchical bayesian models for collaborative tagging systems. In *ICDM '09: Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, pages 728–733, Washington, DC, USA, 2009. IEEE Computer Society.

[30] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, New York, NY, USA, 1998. ACM.

[31] Jonathan Chang. Not-so-latent Dirichlet allocation: collapsed Gibbs sampling using human judgments. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 131–138, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[32] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, NIPS '09, pages 288–296, 2009.

[33] Pimwadee Chaovalit and Lina Zhou. Movie review mining: a comparison between supervised and unsupervised classification approaches. In *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4 - Volume 04*, pages 112.3–, Washington, DC, USA, 2005. IEEE Computer Society.

[34] Harr Chen and David R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 429–436, New York, NY, USA, 2006. ACM.

[35] Judith A. Chevalier and Dina Mayzlin. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345–354, 2006.

[36] Paul-Alexandru Chirita, Stefania Costache, Wolfgang Nejdl, and Siegfried Handschuh. P-tag: large scale automatic generation of personalized annotation tags for the web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 845–854, New York, NY, USA, 2007. ACM.

[37] Yoonjung Choi, Youngho Kim, and Sung-Hyon Myaeng. Domain-Specific Sentiment Analysis Using Contextual Feature Generation. In *TSA '09: Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 37–44, New York, NY, USA, 2009. ACM.

[38] Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information

retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 659–666, New York, NY, USA, 2008. ACM.

[39] Charles L.A. Clarke, Nick Craswell, and Ian Soboroff. Overview of the TREC 2009 Web Track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009*, volume Special Publication 500-278. NIST, 2009.

[40] William S. Cooper. The formalism of probability theory in IR: a foundation or an encumbrance? In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 242–247, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[41] Gianna M. Del Corso, Antonio Gullí, and Francesco Romani. Ranking a stream of news. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 97–106, New York, NY, USA, 2005. ACM.

[42] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proc. of the 40th Anniversary Meeting of the ACL*, 2002.

[43] Yan Dang, Yulei Zhang, and HsinChun Chen. A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews. *IEEE Intelligent Systems*, 25:46–53, 2010. IEEE Computer Society. Los Alamitos, CA, USA.

[44] Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280, New York, NY, USA, 2007. ACM.

[45] Atish Das Sarma, Sreenivas Gollapudi, and Samuel Ieong. Bypass rates: reducing query abandonment using negative inferences. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 177–185, New York, NY, USA, 2008. ACM.

[46] Ritendra Datta, Weina Ge, Jia Li, and James Z. Wang. Toward bridging the annotation-retrieval gap in image search. *Multimedia, IEEE*, 14(3):24–35, July-Sept. 2007.

[47] 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.

[48] Sabine Deligne and Frédéric Bimbot. Language modeling by variable length sequences: theoretical formulation and evaluation of multigrams. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 1, pages 169–172, Los Alamitos, CA, USA, 1995. IEEE Computer Society.

[49] Gianluca Demartini, Claudiu-S Firan, Mihai Georgescu, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. An Architecture for Finding Entities on the Web. In *LA-WEB '09: Proceedings of the 2009 Latin American Web Congress*, pages 230–237, Washington, DC, USA, November 9–11 2009. IEEE Computer Society.

[50] Gianluca Demartini, Claudiu-S Firan, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. A Model for Ranking Entities and Its Application to Wikipedia. In *LA-WEB '08: Proceedings of the 2008 Latin American Web Conference*, pages 29–38, Washington, DC, USA, October 28–30 2008. IEEE Computer Society.

[51] Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. Why finding entities in Wikipedia is difficult, sometimes. *Information Retrieval*, 13(5):534–567, 2010. Springer. Amsterdam, The Netherlands.

[52] Elena Demidova, Peter Fankhauser, Xuan Zhou, and Wolfgang Nejdl. Divq: Diversification for keyword search over structured databases. In *SIGIR '10: Proceedings of the 33rd annual international ACM SIGIR conference on Research and development in information retrieval*, July 19–23 2010.

[53] Kerstin Denecke. Are sentiwordnet scores suited for multi-domain sentiment classification? In Bill Grosky, Frédéric Andrès, and Pit Pichappan, editors, *Fourth IEEE International Conference on Digital Information Management, ICDIM 2009, November 1-4, 2009, University of Michigan, Ann Arbor, Michigan, USA*, pages 33–38. IEEE, 2009.

[54] Pavel A. Dmitriev, Nadav Eiron, Marcus Fontoura, and Eugene J. Shekita. Using annotations in enterprise search. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble, and Michael Dahlin, editors, *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*, pages 811–817, New York, NY, USA, 2006. ACM.

[55] Folke Eisterlehner, Andreas Hotho, and Robert Jäschke, editors. *ECML PKDD Discovery Challenge 2009 (DC09)*, volume 497 of *CEUR-WS.org*, September 2009.

[56] Andrea Esuli and Fabrizio Sebastiani. Determining term subjectivity and term orientation for opinion mining. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April 2006. Association for Computer Linguistics.

[57] Angela Fahrni and Manfred Klenner. Old Wine or Warm Beer: Target-Specific Sentiment Analysis of Adjectives. In *Proc.of the Symposium on Affective Language in Human and Machine, AISB 2008 Convention, 1st-2nd April 2008. University of Aberdeen, Aberdeen, Scotland*, pages 60–63, 2008.

[58] Christiane Fellbaum, editor. *WordNet An Electronic Lexical Database.* The MIT Press, Cambridge, MA; London, May 1998.

[59] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. Building Watson: An overview of the DeepQA project. *AI Magazine*, 31(3):59–79, 2010.

[60] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[61] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric K. Ringger. Pulse: Mining customer opinions from free text. In A. Fazel Famili, Joost N. Kok, José Manuel Peña, Arno Siebes, and A. J. Feelders, editors, *Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis, IDA 2005, Madrid, Spain, September 8-10, 2005, Proceedings*, volume 3646 of *Lecture Notes in Computer Science*, pages 121–132. Springer, 2005.

[62] Nikhil Garg and Ingmar Weber. Personalized, interactive tag recommendation for flickr. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 67–74, New York, NY, USA, 2008. ACM.

[63] Anindya Ghose and Panagiotis G. Ipeirotis. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *Proceedings of the ninth international conference on Electronic commerce*, ICEC '07, pages 303–310, New York, NY, USA, 2007. ACM.

[64] Stefan Gindl, Albert Weichselbraun, and Arno Scharl. Cross-Domain Contextualisation of Sentiment Lexicons. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *19th European Conference on Artificial Intelligence (ECAI)*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 771–776, Lisbon, Portugal, August 2010. IOS Press.

[65] Mark Girolami and Ata Kabán. On an equivalence between plsi and lda. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada*, pages 433–434. ACM, 2003.

[66] Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. Large-Scale Sentiment Analysis for News and Blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007.

[67] Scott A. Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. *CoRR*, abs/cs/0508082, 2005.

[68] Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[69] Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 381–390, New York, NY, USA, 2009. ACM.

[70] Rohitha Goonatilake and Susantha Herath. The volatility of the stock market and news. *International Research Journal of Finance and Economics*, 3(11):53–65, September 2007.

[71] Scott Grant and James R. Cordy. Estimating the optimal number of latent concepts in source code analysis. In *Source Code Analysis and Manipulation, IEEE International Workshop on*, pages 65–74, Los Alamitos, CA, USA, 2010. IEEE Computer Society.

[72] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proc Natl Acad Sci U S A*, 101 Suppl 1:5228–5235, April 2004.

[73] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan O. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, pages 576–587. Morgan Kaufmann, 2004.

[74] Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 362–370, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[75] Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*, pages 211–220. ACM, 2007.

[76] F. Maxwell Harper, Daniel Moy, and Joseph A. Konstan. Facts or Friends?: Distinguishing informational and conversational questions in social Q&A sites. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 759–768. ACM, 2009.

[77] Morgan Harvey, Mark Baillie, Ian Ruthven, and Mark James Carman. Tripartite hidden topic models for personalised tag suggestion. In *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010. Proceedings*, volume 5993 of *Lecture Notes in Computer Science*, pages 432–443. Springer, 2010.

[78] Mark Hepple. Independence and commitment: assumptions for rapid training and execution of rule-based POS taggers. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 278–277, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[79] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004. ACM. New York, NY, USA.

[80] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve web search? In Marc Najork, Andrei Z. Broder, and Soumen Chakrabarti, editors, *Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008*, pages 195–206. ACM, 2008.

[81] Paul Heymann, Daniel Ramage, and Hector Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.

[82] Djoerd Hiemstra. Language models. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 1591–1594. Springer US, 2009.

[83] Thomas Hofmann. Probabilistic latent semantic analysis. In Kathryn B. Laskey and Henri Prade, editors, *UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, July 30-August 1, 1999*, pages 289–296. Morgan Kaufmann, 1999.

[84] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*, pages 411–426, Heidelberg, Germany, June 2006. Springer.

[85] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM Press.

[86] Yang Hu, Mingjing Li, Zhiwei Li, and Wei-Ying Ma. Discovering authoritative news sources and top news stories. In Hwee Tou Ng, Mun-Kew Leong, Min-Yen Kan, and Donghong Ji, editors, *AIRS*, volume 4182 of *Lecture Notes in Computer Science*, pages 230–243. Springer, 2006.

[87] Hal Daumé III and Daniel Marcu. Bayesian query-focused summarization. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. The Association for Computer Linguistics, 2006.

[88] Cisco Systems Inc. Cisco visual networking index: Usage study. `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/Cisco_VNI_Usage_WP.pdf`, October 2010.

[89] VerSign Inc. The domain name industry brief - volume 8 - issue 2. `http://www.verisigninc.com/assets/domain-name-report-may2011.pdf`, May 2011.

[90] Tomoharu Iwata, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. Online multiscale dynamic topic models. In Bharat Rao, Balaji Krishnapuram, Andrew Tomkins, and Qiang Yang, editors, *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, pages 663–672. ACM, 2010.

[91] Anoop Jain, Parag Sarda, and Jayant R. Haritsa. Providing diversity in k-nearest neighbor query results. In *PAKDD'04: Proceedings of the 8th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 404–413, Berlin, Germany, 2004. Springer.

[92] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002. ACM. New York, NY, USA.

[93] Robert Jäschke, Leandro Balby Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 506–514, Heidelberg, Germany, 2007. Springer.

[94] Frederick Jelinek. *Statistical methods for speech recognition.* MIT Press, Cambridge, MA, USA, 1997.

[95] Valentin Jijkoun, Maarten de Rijke, and Wouter Weerkamp. Generating focused topic-specific sentiment lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 585–594, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[96] Karen Sparck Jones. What might be in a summary. *Information Retrieval 93 Von der Modellierung zur Anwendung*, pages 9–26, 1993.

[97] Hiroshi Kanayama and Tetsuya Nasukawa. Fully Automatic Lexicon Expansion for Domain-Oriented Sentiment Analysis. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[98] Hans Mathias Kepplinger and Simone Christine Ehmig. Predicting news decisions. an empirical test of the two-component theory of news selection. *Communications*, 31(1), April 2006.

[99] Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 423–430. Association for Computational Linguistics, 2006.

[100] Ralf Krestel, Sabine Bergler, and René Witte. A Belief Revision Approach to Textual Entailment Recognition. In *Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA, November 17-19 2008. National Institute of Standards and Technology (NIST).

[101] Ralf Krestel, Sabine Bergler, and René Witte. Minding the source: Automatic tagging of reported speech in newspaper articles. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, pages 2823–2828, Marrakech, Morocco, May 28-30 2008. European Language Resources Association (ELRA).

[102] Ralf Krestel, Sabine Bergler, and René Witte. Believe It or Not: Solving the TAC 2009 Textual Entailment Tasks through an Artificial Believer System. In *Text Analysis Conference (TAC)*. National Institute of Standards and Technology (NIST), November 16–17 2009.

[103] Ralf Krestel and Ling Chen. Using Co-occurence of Tags and Resources to Identify Spammers. In *Proceedings of the ECML/PKDD 2008 Discovery Challenge*, Antwerp, Belgium, September 2008.

[104] Ralf Krestel and Ling Chen. The Art of Tagging: Measuring the Quality of Tags. In *ASWC '08: Proceedings of the 3rd Asian Semantic Web Conference on The Semantic Web*, volume 5367 of *Lecture Notes in Computer Science*, pages 257–271, Berlin, Heidelberg, February 2–5 2009. Springer-Verlag.

[105] Ralf Krestel, Gianluca Demartini, and Eelco Herder. Visual Interfaces for Stimulating Exploratory Search. In *Proceedings of the 2011 Joint International Conference on Digital Libraries (JCDL 2011)*, pages 393–394. ACM, June 13–17 2011.

[106] Ralf Krestel and Nima Dokoohaki. Diversifying Product Review Rankings: Getting the Full Picture. In *WI-IAT '11: Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Washington, DC, USA, August 22–27 2011. IEEE Computer Society. BEST PAPER AWARD.

[107] Ralf Krestel and Peter Fankhauser. Tag recommendation using probabilistic topic models. In *ECML/PKDD Discovery Challenge (DC'09), Workshop at ECML/PKDD 2009*, pages 131–141, September 7th 2009.

[108] Ralf Krestel and Peter Fankhauser. Language Models & Topic Models for Personalizing Tag Recommendation. In *WI-IAT '10: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 82–89, Washington, DC, USA, August 31–September 3 2010. IEEE Computer Society.

[109] Ralf Krestel and Peter Fankhauser. Personalized topic-based tag recommendation. *Neurocomputing*, 76(1):61–70, 2012. Elsevier. Amsterdam, Netherlands.

[110] Ralf Krestel and Peter Fankhauser. Web Search Result Diversification by Reranking. *Information Retrieval*, 2012. Springer. Amsterdam, Netherlands. (in press).

[111] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent Dirichlet Allocation for Tag Recommendation. In *Proceedings of the 2009 ACM Conference on Recommender Systems (RecSys 2009)*, pages 61–68, New York City, New York, USA, October 23–25 2009. ACM.

[112] Ralf Krestel and Bhaskar Mehta. Predicting News Story Importance using Language Features. In *WI-IAT '08: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 683–689, Washington, DC, USA, December 9–12 2008. IEEE Computer Society.

[113] Ralf Krestel and Bhaskar Mehta. Learning the Importance of Latent Topics to Discover Highly Influential News Items. In *KI 2010: Advances in Artificial Intelligence, 33rd Annual German Conference on AI, Karlsruhe, Germany, September 21-24, 2010. Proceedings*, volume 6359 of *Lecture Notes in Computer Science*, pages 211–218, Berlin, Heidelberg, September 21–24 2010. Springer. BEST PAPER AWARD.

[114] Ralf Krestel, René Witte, and Sabine Bergler. Creating a Fuzzy Believer to Model Human Newspaper Readers. In Z. Kobti and D. Wu, editors, *Proceedings of the 20th Canadian Conference on Artificial Intelligence (CAI 2007)*, LNAI 4509, pages 489–501, Montréal, Québec, Canada, May 28–30 2007. Springer.

[115] Ralf Krestel, René Witte, and Sabine Bergler. Fuzzy Set Theory-Based Belief Processing for Natural Language Texts. In *AAAI'07: Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 1878–1879. AAAI Press, July 22–26 2007.

[116] Ralf Krestel, René Witte, and Sabine Bergler. Processing of Beliefs extracted from Reported Speech in Newspaper Articles. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007)*, Borovets, Bulgaria, September 27–29 2007.

[117] Ralf Krestel, René Witte, and Sabine Bergler. Predicate-Argument EXtractor (PAX). In *Proceedings of the First Workshop on New Challenges for NLP Frameworks co-located with LREC 2010*, pages 51–54. European Language Resources Association (ELRA), May 22nd 2010.

[118] Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Sentiment analysis with global topics and local dependency. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010.

[119] Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 577–584, New York, NY, USA, 2006. ACM.

[120] Marek Lipczak and Evangelos Milios. The impact of resource title on tags in collaborative tagging systems. In *HT '10: Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pages 179–188, New York, NY, USA, 2010. ACM.

[121] Walter Lippmann. *Public Opinion.* Harcourt, Brace and Company New York, 1922.

[122] Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. ARSA: A Sentiment-Aware Model for Predicting Sales Performance Using Blogs. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 607–614, New York, NY, USA, 2007. ACM.

[123] Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. Modeling and predicting the helpfulness of online reviews. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 443–452, Washington, DC, USA, 2008. IEEE Computer Society.

[124] Christop Lofi and Ralf Krestel. iParticipate: Automatic Tweet Generation from Local Government Data. In *Proceedings of the 17th International Conference on Database Systems for Advanced Applications (DASFAA).* Springer, April 15–18 2012. (in press).

[125] Yue Lu, ChengXiang Zhai, and Neel Sundaresan. Rated aspect summarization of short comments. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 131–140, New York, NY, USA, 2009. ACM.

[126] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, Cambridge, UK, July 2008.

[127] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing.* MIT Press, Cambridge, MA, USA, 1999.

[128] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. Blackwell Publishing for the American Finance Association.

[129] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT 2006, Proceedings of the 17th ACM Conference on Hypertext and Hypermedia, August 22-25, 2006, Odense, Denmark*, pages 31–40, New York, NY, USA, 2006. ACM.

[130] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. Yale: Rapid prototyping for complex data mining tasks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, NY, USA, August 2006. ACM.

[131] David R. H. Miller, Tim Leek, and Richard M. Schwartz. A hidden markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 214–221, New York, NY, USA, 1999. ACM.

[132] Thomas P. Minka and John D. Lafferty. Expectation-propogation for the generative aspect model. In Adnan Darwiche and Nir Friedman, editors, *UAI '02, Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence, University of Alberta, Edmonton, Alberta, Canada, August 1-4, 2002*, pages 352–359. Morgan Kaufmann, 2002.

[133] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM.

[134] Dunja Mladenić, Janez Brank, Marko Grobelnik, and Natasa Milic-Frayling. Feature Selection Using Linear Classifier Weights: Interaction with Classification Models. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 234–241, New York, NY, USA, 2004. ACM.

[135] David Newman, Timothy Baldwin, Lawrence Cavedon, Eric Huang, Sarvnaz Karimi, David Martinez, Falk Scholer, and Justin Zobel. Invited paper: Visualizing search results and document collections using topic maps. *Web Semant.*, 8:169–175, July 2010. Elsevier Science Publishers B. V. Amsterdam, The Netherlands.

[136] 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.

[137] Scott Nowson. Scary Films Good, Scary Flights Bad: Topic Driven Feature Selection for Classification of Sentiment. In *TSA '09: Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 17–24, New York, NY, USA, 2009. ACM.

[138] Michael P. O'Mahony, Pádraig Cunningham, and Barry Smyth. An assessment of machine learning techniques for review recommendation. In *Proceedings of the 20th Irish conference on Artificial intelligence and cognitive science*, AICS'09, pages 241–250, Berlin, Heidelberg, 2010. Springer-Verlag.

[139] Michael P. O'Mahony and Barry Smyth. Learning to recommend helpful hotel reviews. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 305–308, New York, NY, USA, 2009. ACM.

[140] Einar Østgaard. Factors influencing the flow of news. *Journal of Peace Research*, 2:39–63, 1965.

[141] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[142] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-Domain Sentiment Classification via Spectral Feature Alignment. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 751–760, New York, NY, USA, 2010. ACM.

[143] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2007.

[144] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[145] Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 1-3, 1998, Seattle, Washington*, pages 159–168. ACM Press, 1998.

[146] Michael J. Paul and Roxana Girju. A two-dimensional topic-aspect model for discovering multi-faceted topics. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010.

[147] Jay M. Ponte and W. 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*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[148] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 339–346. Association for Computational Linguistics, 2005.

[149] Foster Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconson, USA, July 24-27, 1998*, pages 445–453, San Francisco, CA, USA, 1998. Morgan Kaufmann.

[150] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Expanding Domain Sentiment Lexicon Through Double Propagation. In *IJCAI'09: Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1199–1204, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[151] Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. The Bag-of-Opinions Method for Review Rating Prediction from Sparse Text Patterns. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 913–921, Beijing, China, August 2010. Coling 2010 Organizing Committee.

[152] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann, San Francisco, CA, USA, 1993.

[153] Filip Radlinksi, Paul N. Bennet, Ben Carterette, and Thorsten Joachims. Redundancy, diversity and interdependent document relevance - workshop report. *ACM SIGIR Forum*, 43(2), December 2009.

[154] Filip Radlinski and Susan Dumais. Improving personalized web search using result diversification. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 691–692, New York, NY, USA, 2006. ACM.

[155] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 784–791, New York, NY, USA, 2008. ACM.

[156] Davood Rafiei, Krishna Bharat, and Anand Shukla. Diversifying web search results. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 781–790, New York, NY, USA, 2010. ACM.

[157] Daniel Ramage, Susan T. Dumais, and Daniel J. Liebling. Characterizing microblogs with topic models. In William W. Cohen and Samuel Gosling, editors, *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*. The AAAI Press, 2010.

[158] 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.

[159] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90, New York, NY, USA, 2010. ACM.

[160] Stephen E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, pages 294–304, 1977.

[161] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

[162] Gerard Salton, Amit Singhal, Chris Buckley, and Mandar Mitra. Automatic text decomposition using text segments and text themes. In *Proceedings of the the seventh ACM conference on Hypertext HYPERTEXT 96*, pages 53–65. ACM Press, 1996.

[163] Gerard Salton, Andrew Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, November 1975. ACM. New York, NY, USA.

[164] Celina Santamaría, Julio Gonzalo, and Javier Artiles. Wikipedia as sense inventory to improve diversity in web search results. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1357–1366, Uppsala, Sweden, July 2010. ACL.

[165] Ralf Schenkel, Tom Crecelius, Mouna Kacimi, Sebastian Michel, Thomas Neumann, Josiane X. Parreira, and Gerhard Weikum. Efficient top-k querying over social-tagging networks. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 523–530, New York, NY, USA, 2008. ACM.

[166] Dominik Schnitzer, Arthur Flexer, and Gerhard Widmer. A filter-and-refine indexing method for fast similarity search in millions of music tracks. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR'09)*, 2009.

[167] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin D. Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23-25, 2008*, pages 259–266, New York, NY, USA, 2008. ACM.

[168] Xiance Si and Maosong Sun. Tag-lda for scalable real-time tag recommendation. *Journal of Computational Information Systems*, 6(1):23–31, January 2009.

[169] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 327–336, New York, NY, USA, 2008. ACM.

[170] Fei Song and W. Bruce Croft. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*, CIKM '99, pages 316–321, New York, NY, USA, 1999. ACM.

[171] Yang Song, Ziming Zhuang, Huajing Li, Qiankun Zhao, Jia Li, Wang-Chien Lee, and C. Lee Giles. Real-time automatic tag recommendation. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 515–522, New York, NY, USA, 2008. ACM.

[172] Mark Steyvers and Thomas L. Griffiths. Probabilistic topic models. In T. Landauer, D. Mcnamara, S. Dennis, and W. Kintsch, editors, *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, 2007.

[173] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 43–50, New York, NY, USA, 2008. ACM.

[174] Yee W. Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. *Advances in Neural Information Processing Systems 17*, pages 1385–1392, 2005.

[175] Yee Whye Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In Bernhard Schölkopf, John C. Platt, and Thomas Hoffman, editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 1353–1360. MIT Press, 2007.

[176] Matt Thomas, Bo Pang, and Lillian Lee. Get Out the Vote: Determining Support or Opposition from Congressional Floor-Debate Transcripts. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[177] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 801–810, Washington, DC, USA, 2007. IEEE Computer Society.

[178] Nava Tintarev and Judith Masthoff. The effectiveness of personalized movie explanations: An experiment using commercial meta-data. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 204–213. Springer, 2008.

[179] Ivan Titov and Ryan T. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 308–316. The Association for Computer Linguistics, 2008.

[180] Ivan Titov and Ryan T. McDonald. Modeling Online Reviews with Multi-Grain Topic Models. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, pages 111–120. ACM, 2008.

[181] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *In Proceedings of HLT-NAACL 2003*, pages 252–259, 2003.

[182] Peter D. Turney. Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[183] Peter D. Turney and Michael L. Littman. Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Trans. Inf. Syst.*, 21(4):315–346, 2003. ACM. New York, NY, USA.

[184] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 2000.

[185] Erik Vee, Utkarsh Srivastava, Jayavel Shanmugasundaram, Prashant Bhat, and Sihem Amer-Yahia. Efficient computation of diverse query results. In *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, Cancún, México*, pages 228–236. IEEE, 2008.

[186] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David M. Mimno. Evaluation methods for topic models. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, page 139. ACM, 2009.

[187] Chong Wang, David M. Blei, and Fei-Fei Li. Simultaneous image classification and annotation. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1903–1910. IEEE, 2009.

[188] Jun Wang and Jianhan Zhu. Portfolio theory of information retrieval. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 115–122, New York, NY, USA, 2009. ACM.

[189] Wouter Weerkamp and Maarten de Rijke. Credibility improves topical blog post retrieval. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 923–931. The Association for Computer Linguistics, 2008.

[190] Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA, 2006. ACM.

[191] Robert Wetzker, Carsten Zimmermann, and Christian Bauckhage. Analyzing Social Bookmarking Systems: A del.icio.us Cookbook. In *Proceedings of the ECAI 2008 Mining Social Data Workshop*, pages 26–30, 2008.

[192] Robert Wetzker, Carsten Zimmermann, Christian Bauckhage, and Sahin Albayrak. I tag, you tag: translating tags for advanced user models. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 71–80, New York, NY, USA, 2010. ACM.

[193] Casey Whitelaw, Navendu Garg, and Shlomo Argamon. Using Appraisal Groups for Sentiment Analysis. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 625–631, New York, NY, USA, 2005. ACM.

[194] Janyce Wiebe and Ellen Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of CICLing*, pages 486–497. Springer, 2005.

[195] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433, 2009.

[196] René Witte, Thomas Gitzinger, Thomas Kappler, and Ralf Krestel. A Semantic Wiki Approach to Cultural Heritage Data Management. In *Language Technology for Cultural Heritage Data (LaTeCH 2008) Workshop at LREC 2008*, Marrakech, Morocco, June 1st 2008.

[197] René Witte, Thomas Kappler, Ralf Krestel, and Peter Lockemann. Integrating Wiki Systems, Natural Language Processing, and Semantic Technologies for Cultural Heritage Data Management. In Caroline Sporleder, Antal van den Bosch, and Kalliopi A. Zervanou, editors, *Language Technology for Cultural Heritage*, Theory and Applications of Natural Language Processing, pages 213–230. Springer Berlin Heidelberg, 2011.

[198] René Witte and Ralf Krestel. Semantic Content Access using Domain-Independent NLP Ontologies. In *Natural Language Processing and Information Systems, 15th International Conference on Applications of Natural Language to Information Systems, NLDB 2010, Cardiff, UK, June 23-25, 2010. Proceedings*, volume 6177 of *Lecture Notes in Computer Science*, pages 36–47, Berlin, Heidelberg, June 23–25 2010. Springer.

[199] René Witte, Ralf Krestel, and Sabine Bergler. Generating Update Summaries for DUC 2007. In *Proceedings of Document Understanding Workshop (DUC)*, Rochester, NY, USA, April 26–27 2007.

[200] René Witte, Ralf Krestel, Thomas Kappler, and Peter Lockemann. Converting a Historical Encyclopedia of Architecture into a Semantic Knowledge Base. *IEEE Intelligent Systems*, 25(1):58–67, 2010. IEEE Computer Society. Los Alamitos, CA, USA.

[201] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, San Francisco, CA, USA, second edition, June 2005.

[202] Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the semantic web: Collaborative tag suggestions. In *Proceedings of Collaborative Web Tagging Workshop at 15th International World Wide Web Conference*, 2006.

[203] Changhua Yang, Kevin Hsin-Yih Lin, and Hsin-Hsi Chen. Building emotion lexicon from weblog corpora. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 133–136, Morristown, NJ, USA, 2007. Association for Computational Linguistics.

[204] Yiming Yang and Jan O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[205] Yang Yang Song, Lu Zhang, and C. Lee Giles. Automatic Tag Recommendation Algorithms for Social Recommender Systems. *ACM Transactions on the Web*, 2010.

[206] Jinyi Yao, Jue Wang, Zhiwei Li, Mingjing Li, and Wei-Ying Ma. Ranking web news via homepage visual layout and cross-site voting. In Mounia Lalmas, Andy MacFarlane, Stefan M. Rüger, Anastasios Tombros, Theodora Tsikrika, and Alexei Yavlinsky, editors, *ECIR*, volume 3936 of *Lecture Notes in Computer Science*, pages 131–142. Springer, 2006.

[207] ChengXiang Zhai, William W. Cohen, and John Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 10–17, New York, NY, USA, 2003. ACM.

[208] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 334–342, New York, NY, USA, 2001. ACM.

[209] ChengXiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004. ACM. New York, NY, USA.

[210] ChengXiang Zhai and John Lafferty. A risk minimization framework for information retrieval. *Inf. Process. Manage.*, 42(1):31–55, 2006. Pergamon Press, Inc. Tarrytown, NY, USA.

[211] Zhaohui Zheng, Xiaoyun Wu, and Rohini Srihari. Feature selection for text categorization on imbalanced data. *SIGKDD Explor. Newsl.*, 6(1):80–89, 2004. ACM. New York, NY, USA.

[212] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management - CIKM '06*, page 43. ACM Press, 2006.

[213] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 22–32, New York, NY, USA, 2005. ACM.

ACADEMIC VITAE

Educational Background

2007–2011	Researcher at L3S Research Center and Leibniz Universität Hannover, Germany
2001–2007	Diplom, Computer Science, Universität Karlsruhe (TH), Germany
2006, 2007	Visiting Scholor, Research Stay, Concordia University, Montréal, Canada
1991–2000	Abitur, Hans Furler Gymnasium, Oberkirch, Baden-Württemberg, Germany

Publications

Book Chapters

- René Witte, Thomas Kappler, Ralf Krestel, and Peter Lockemann. Integrating Wiki Systems, Natural Language Processing, and Semantic Technologies for Cultural Heritage Data Management. In Caroline Sporleder, Antal van den Bosch, and Kalliopi A. Zervanou, editors, *Language Technology for Cultural Heritage*, Theory and Applications of Natural Language Processing, pages 213–230. Springer Berlin Heidelberg, 2011

Journal Articles

- Ralf Krestel and Peter Fankhauser. Web Search Result Diversification by Reranking. *Information Retrieval*, 2012. Springer. Amsterdam, Netherlands. (in press)

- Ralf Krestel and Peter Fankhauser. Personalized topic-based tag recommendation. *Neurocomputing*, 76(1):61–70, 2012. Elsevier. Amsterdam, Netherlands

- René Witte, Ralf Krestel, Thomas Kappler, and Peter Lockemann. Converting a Historical Encyclopedia of Architecture into a Semantic Knowledge Base. *IEEE Intelligent Systems*, 25(1):58–67, 2010. IEEE Computer Society. Los Alamitos, CA, USA

- Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. Why finding entities in Wikipedia is difficult, sometimes. *Information Retrieval*, 13(5):534–567, 2010. Springer. Amsterdam, The Netherlands

Conference Papers

- Ralf Krestel and Nima Dokoohaki. Diversifying Product Review Rankings: Getting the Full Picture. In *WI-IAT '11: Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Washington, DC, USA, August 22–27 2011. IEEE Computer Society. BEST PAPER AWARD

- Ralf Krestel and Bhaskar Mehta. Learning the Importance of Latent Topics to Discover Highly Influential News Items. In *KI 2010: Advances in Artificial Intelligence, 33rd Annual German Conference on AI, Karlsruhe, Germany, September 21-24, 2010. Proceedings*, volume 6359 of *Lecture Notes in Computer Science*, pages 211–218, Berlin, Heidelberg, September 21–24 2010. Springer. BEST PAPER AWARD

- Bodo Billerbeck, Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, and Ralf Krestel. Ranking Entities Using Web Search Query Logs. In *Research and Advanced Technology for Digital Libraries, 14th European Conference, ECDL 2010, Glasgow, UK, September 6-10, 2010. Proceedings*, volume 6273 of *Lecture Notes in Computer Science*, pages 273–281, Berlin, Heidelberg, September 6–10 2010. Springer

- Ralf Krestel and Peter Fankhauser. Language Models & Topic Models for Personalizing Tag Recommendation. In *WI-IAT '10: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 82–89, Washington, DC, USA, August 31–September 3 2010. IEEE Computer Society

- René Witte and Ralf Krestel. Semantic Content Access using Domain-Independent NLP Ontologies. In *Natural Language Processing and Information Systems, 15th International Conference on Applications of Natural Language to Information Systems, NLDB 2010, Cardiff, UK, June 23-25, 2010. Proceedings*, volume 6177 of *Lecture Notes in Computer Science*, pages 36–47, Berlin, Heidelberg, June 23–25 2010. Springer

- Gianluca Demartini, Claudiu-S Firan, Mihai Georgescu, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. An Architecture for Finding Entities on the Web. In *LA-WEB '09: Proceedings of the 2009 Latin American Web Congress*, pages 230–237, Washington, DC, USA, November 9–11 2009. IEEE Computer Society

- Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent Dirichlet Allocation for Tag Recommendation. In *Proceedings of the 2009 ACM Conference on Recommender Systems (RecSys 2009)*, pages 61–68, New York City, New York, USA, October 23–25 2009. ACM

- Ralf Krestel and Bhaskar Mehta. Predicting News Story Importance using Language Features. In *WI-IAT '08: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 683–689, Washington, DC, USA, December 9–12 2008. IEEE Computer Society

- Ralf Krestel and Ling Chen. The Art of Tagging: Measuring the Quality of Tags. In *ASWC '08: Proceedings of the 3rd Asian Semantic Web Conference on The Semantic Web*, volume 5367 of *Lecture Notes in Computer Science*, pages 257–271, Berlin, Heidelberg, February 2–5 2009. Springer-Verlag

- Gianluca Demartini, Claudiu-S Firan, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. A Model for Ranking Entities and Its Application to Wikipedia. In *LA-WEB '08: Proceedings of the 2008 Latin American Web Conference*, pages 29–38, Washington, DC, USA, October 28–30 2008. IEEE Computer Society

- Ralf Krestel, Sabine Bergler, and René Witte. Minding the source: Automatic tagging of reported speech in newspaper articles. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, pages 2823–2828, Marrakech, Morocco, May 28-30 2008. European Language Resources Association (ELRA)

- Ralf Krestel, René Witte, and Sabine Bergler. Creating a Fuzzy Believer to Model Human Newspaper Readers. In Z. Kobti and D. Wu, editors, *Proceedings of the 20th Canadian Conference on Artificial Intelligence (CAI 2007)*, LNAI 4509, pages 489–501, Montréal, Québec, Canada, May 28–30 2007. Springer

- Ralf Krestel, René Witte, and Sabine Bergler. Processing of Beliefs extracted from Reported Speech in Newspaper Articles. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007)*, Borovets, Bulgaria, September 27–29 2007

Workshop Papers

- Ralf Krestel, René Witte, and Sabine Bergler. Predicate-Argument EXtractor (PAX). In *Proceedings of the First Workshop on New Challenges for NLP Frameworks co-located with LREC 2010*, pages 51–54. European Language Resources Association (ELRA), May 22nd 2010

- Ralf Krestel, Sabine Bergler, and René Witte. Believe It or Not: Solving the TAC 2009 Textual Entailment Tasks through an Artificial Believer System. In *Text Analysis Conference (TAC)*. National Institute of Standards and Technology (NIST), November 16–17 2009

- Ralf Krestel and Peter Fankhauser. Tag recommendation using probabilistic topic models. In *ECML/PKDD Discovery Challenge (DC'09), Workshop at ECML/PKDD 2009*, pages 131–141, September 7th 2009

- René Witte, Thomas Gitzinger, Thomas Kappler, and Ralf Krestel. A Semantic Wiki Approach to Cultural Heritage Data Management. In *Language Technology for Cultural Heritage Data (LaTeCH 2008) Workshop at LREC 2008*, Marrakech, Morocco, June 1st 2008

- Ralf Krestel and Ling Chen. Using Co-occurence of Tags and Resources to Identify Spammers. In *Proceedings of the ECML/PKDD 2008 Discovery Challenge*, Antwerp, Belgium, September 2008

- Ralf Krestel, Sabine Bergler, and René Witte. A Belief Revision Approach to Textual Entailment Recognition. In *Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA, November 17-19 2008. National Institute of Standards and Technology (NIST)

- René Witte, Ralf Krestel, and Sabine Bergler. Generating Update Summaries for DUC 2007. In *Proceedings of Document Understanding Workshop (DUC)*, Rochester, NY, USA, April 26–27 2007

Posters

- Christop Lofi and Ralf Krestel. iParticipate: Automatic Tweet Generation from Local Government Data. In *Proceedings of the 17th International Conference on Database Systems for Advanced Applications (DASFAA)*. Springer, April 15–18 2012. (in press)

- Ralf Krestel, Gianluca Demartini, and Eelco Herder. Visual Interfaces for Stimulating Exploratory Search. In *Proceedings of the 2011 Joint International Conference on Digital Libraries (JCDL 2011)*, pages 393–394. ACM, June 13–17 2011

- Bodo Billerbeck, Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, and Ralf Krestel. Exploiting click-through data for entity retrieval. In *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 168–169. ACM, July 19–23 2010

- Ralf Krestel, René Witte, and Sabine Bergler. Fuzzy Set Theory-Based Belief Processing for Natural Language Texts. In *AAAI'07: Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 1878–1879. AAAI Press, July 22–26 2007