ANALYZING AND IMPROVING DIVERSIFICATION, PRIVACY, AND INFORMATION MANAGEMENT ON THE WEB

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

DOKTOR DER NATURWISSENSCHAFTEN

Dr. rer. nat.

genehmigte Dissertation
von

Dipl.-Math. Kaweh Djafari Naini

geboren am 11. Mai 1984, in Teheran, Iran

Hannover, Deutschland, 2019

Referent: Prof. Dr. techn. Wolfgang Nejdl
Korreferent: Assoc. Prof. Dr. Ismail Sengor Altingovde
Tag der Promotion: 13.11.2018

ABSTRACT

Today, the World Wide Web has become the main source and medium for people to access, share, and manage information. Since user expectations towards all three types of functionalities are high and information volumes are growing very fast, modern web applications are exposed to new challenges by supporting the users in their daily and long-term interactions on the web. In this thesis, we contribute to the following core challenges related to the aforementioned functionalities.

Diversification for improving information access - in Web search engines the user can access information by submitting a query that returns a set of search results. Web search queries often contain only a few terms, and can be ambiguous, which is a core issue for retrieval systems. For instance, modern search engines extract a large amount of additional features for building a sophisticated ranking model. Further, recent studies on web search results diversification show that retrieval effectiveness for ambiguous queries can be considerably improved by diversifying the search results. In this thesis, we present two approaches for improving retrieval effectiveness and efficiency. First, we present an efficient and scalable algorithm for web search results diversification for large-scale retrieval systems. Second, we present an approach for feature selection in learning-to-rank.

Privacy issues and communication practices through information sharing - social networks allow the user to share information to a wider audience or communicate within specific groups. Understanding the users' motivation and behavior in social networks is crucial for supporting the users' needs, e.g. by suggesting relevant resources or creating new services. In recent years, the increasing amount of personal information shared in social networks has exposed users to risks of endangering their privacy. Popular social networks often allow the user to manually control the privacy settings of social content before it is shared. However, existing functionalities for privacy settings are often restricted and very time consuming for the user. In this thesis, we present an approach for predicting privacy settings of the user. Furthermore, we present an in-depth study of social and professional networks for identifying communication practices for different types of users with different skills and expertise.

Personalized and long-term information management for social content - the information flood in social media makes it is nearly impossible for users to manually manage their social media posts over several years. Approaches for summarizing and aggregating of social media postings face the challenge to identify information from the past that is still relevant in the future, i.e., for reminiscence or inclusion into a summary. In this thesis, we conduct user evaluation studies to better capture the users' expectation towards information retention. Next, we extract various of features from social media posts, profile and network of the users. Finally, we build general and personalized ranking models for retention, and present a set of seed features which perform best of identifying memorable posts.

The approaches in this thesis are compared to existing baselines and state of the art approaches from related work.

Keywords: *web search results diversification, scalability and efficiency in web search, letor, feature selection, privacy prediction, social network analysis, social media summary*

ZUSAMMENFASSUNG

Heutzutage ist das World Wide Web die wichtigste Quelle zur Informationsbeschaffung, zum Informationsaustausch und der Verwaltung von Informationen. Da die Erwartungen der Nutzer bezüglich der drei vorgenannten Funktionalitäten hoch sind und das Informationsvolumen sehr schnell wächst, sind moderne Webanwendungen stets neuen Herausforderungen ausgesetzt, um die Nutzer bei ihren täglichen und langfristigen Interaktionen im Web unterstützen zu können. In dieser Dissertation tragen wir zu den folgenden zentralen Herausforderungen bei, die sich auf die oben genannte Funktionalitäten beziehen.

Diversifizierung zur Verbesserung des Informationszugangs - In Web-Suchmaschinen kann der Nutzer auf Informationen zugreifen, indem er eine Abfrage sendet, die eine Reihe von Suchergebnissen zurückliefert. Web-Suchanfragen enthalten oft nur wenige Begriffe und können mehrdeutig sein, was für Retrieval-Systeme ein Kernproblem darstellt. Zum Beispiel extrahieren moderne Suchmaschinen eine große Menge zusätzlicher Merkmale, um ein ausgeklügeltes Ranking-Modell zu erstellen. Darüber hinaus zeigen neuere Studien zur Diversifizierung von Websuchergebnissen, dass die Retrieval-Effektivität für mehrdeutige Abfragen durch Diversifizierung der Suchergebnisse erheblich verbessert werden kann. In dieser Arbeit präsentieren wir zwei Ansätze zur Verbesserung der Retrieval-Effektivität und -Effizienz. Zunächst stellen wir einen effizienten und skalierbaren Algorithmus für die Diversifizierung von Web-Suchergebnissen für große Retrieval-Systeme vor. Zweitens präsentieren wir einen Ansatz für die Merkmalauswahl im Learning-to-Rank.

Datenschutzprobleme und Kommunikationspraktiken durch Informationsaustausch - Soziale Netzwerke ermöglichen dem Nutzer, Informationen an ein breites Publikum weiterzugeben oder innerhalb bestimmter Gruppen zu kommunizieren. Um die Nutzer unterstützen zu können, ist es erforderlich, ihre Motivation und ihr Verhalten in sozialen Netzwerken zu verstehen, indem z.B. relevante Ressourcen vorgeschlagen oder neue Dienste angeboten werden. Indem die Nutzer in den letzten Jahren zunehmend persönliche Informationen in den sozialen Netzwerken teilen, setzen sie sich dem Risiko aus, ihre Privatsphäre zu gefährden. Beliebte soziale Netzwerke ermöglichen es dem Nutzer häufig, die Datenschutzeinstellungen von sozialen Inhalten vor der Freigabe manuell zu steuern. Die zum Schutz der Privatsphäre vorhandenen Funktionen sind jedoch oft eingeschränkt und für den Nutzer sehr zeitaufwendig. In dieser Arbeit präsentieren wir einen Ansatz zur Vorhersage von Datenschutzeinstellungen des Nutzers. Darüber hinaus stellen wir eine eingehende Studie über soziale und berufliche Netzwerke zur Identifizierung von Kommunikationspraktiken für verschiedene Arten von Nutzern mit unterschiedlichen Fähigkeiten und Kenntnissen vor.

Personalisiertes und langfristiges Informationsmanagement für soziale Inhalte - Die Informationsflut in den sozialen Medien macht es den Nutzern nahezu unmöglich, ihre Social-Media-Beiträge über mehrere Jahre hinweg manuell zu verwalten. Lösungsansätze zum Sammeln von Social-Media-Beiträge stehen vor der Herausforderung, Informationen aus der Vergangenheit zu identifizieren, die in der Zukunft für den Nutzer denkwürdig sind und für die Erstellung von Zusammenfassungen in Frage kommen. In dieser Arbeit führen wir Nutzerbewertungsstudien durch, um die Erwartungen der Nutzer an die Informationserhaltung besser zu erfassen. Als nächstes extrahieren wir verschiedene Merkmale aus Social-Media-Beiträgen sowie aus Profilen und Netzwerken der Nutzer. Schließlich erstellen wir allgemeine und personalisierte Ranking-Modelle für die Aufbewahrung von Beiträgen. Zusätzlich stellen wir eine Reihe von Kernfunktionen vor, die am besten geeignet sind, denkwürdige Beiträge zu identifizieren.

Die Ansätze in dieser Arbeit werden mit bestehenden Baselines und State-of-the-Art

Ansätzen aus verwandten Arbeiten verglichen.

Schlagwörter: *Diversifizierung von Web Suchergebnissen, Skalierbarkeit und Effizienz in der Websuche, letor, Merkmalsauswahl, Privatsphäre Vorhersage, soziale Netzwerkanalyse, Zusammenfassung in sozialen Medien*

FOREWORD

The contributions presented in this thesis have previously appeared in several conference and journal papers as well as one book chapter published in the course of this PhD program:

The contributions in Chapter 3 are published in:

- Kaweh Djafari Naini, Ismail Sengor Altingovde, and Wolf Siberski. Scalable and efficient web search result diversification. *ACM Transactions on the Web*, *TWEB*, 10(3):15:1-15:30, August 2016.

The contributions in Chapter 4 are published in:

- Kaweh Djafari Naini and Ismail Sengor Altingovde. Exploiting result diversification methods for feature selection in learning to rank. In *Proceedings of the 36th European Conference on Information Retrieval*, *ECIR'14*, pages 455-461, 2014.

The contributions in Chapter 5 are published in:

- Kaweh Djafari Naini, Ismail Sengor Altingovde, Ricardo Kawase, Eelco Herder, and Claudia Niederée. Analyzing and Predicting Privacy Settings in the Social Web. In *Proceedings of the 23rd International Conference on User Modeling, Adaptation and Personalization*, *UMAP'15*, pages 104-117, Dublin, Ireland, June 29 - July 3, 2015.

The contributions in Chapter 6 are published in:

- Sergiu Chelaru, Eelco Herder, Kaweh Djafari Naini, and Patrick Siehndel. Recognizing skill networks and their specific communication and connection practices. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, *HT'14*, pages 13-23, Santiago, Chile, 2014.

The contributions in Chapter 7 are published in:

- Kaweh Djafari Naini, Ricardo Kawase, Nattiya Kanhabua, Claudia Niederée, and Ismail Sengor Altingovde. Those were the days: learning to rank social media posts for reminiscence. *Information Retrieval Journal*, pages 1-29, 2018.

- Kaweh Djafari Naini, Ricardo Kawase, Nattiya Kanhabua, and Claudia Niederée. Characterizing high-impact features for content retention in social web applications. In *23rd International World Wide Web Conference, WWW'14*, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume, pages 559-560, 2014.

- Claudia Niederée, Nattiya Kanhabua, Tuan Tran, and Kaweh Djafari Naini. Preservation Value and Managed Forgetting. In *Book of Personal Multimedia Preservation*, 2018, pages 101-129.

Other publications that I have co-authored during my PhD are listed below:

- Asmelash Teka Hadgu, Kaweh Djafari Naini, and Claudia Niederée. Welcome or not-welcome: Reactions to refugee situation on social media. *CoRR*, arXiv, abs/1610.02358, 2016.

- Ujwal Gadiraju, Kaweh Djafari Naini, Andrea Ceroni, Mihai Georgescu, Dang Duc Pham, and Marco Fisichella. Wikipevent: Temporal event data for the semantic web. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014*, Riva del Garda, Italy, October 21, 2014, pages 125-128.

- Andrea Ceroni, Mihai Georgescu, Ujwal Gadiraju, Kaweh Djafari Naini, and Marco Fisichella. Information evolution in wikipedia. In *Proceedings of The International Symposium on Open Collaboration, OpenSym 2014*, Berlin, Germany, August 27 - 29, 2014, pages 24:1-24:10.

- Tuan A. Tran, Andrea Ceroni, Mihai Georgescu, Kaweh Djafari Naini, and Marco Fisichella. Wikipevent: Leveraging wikipedia edit history for event detection. In *Proceedings of the 15th International Conference Web Information Systems Engineering, WISE 2014*, Thessaloniki, Greece, October 12-14, 2014, Proceedings, Part II, pages 90-108.

- Ernesto Diaz-Aviles, Patrick Siehndel, and Kaweh Djafari Naini. Exploiting social #-tagging behavior in twitter for information filtering and recommendation. In *Proceedings of The Twentieth Text REtrieval Conference, TREC 2011*, Gaithersburg, Maryland, USA, November 15-18, 2011.

Contents

List of Figures

List of Tables

1 Motivation

Nowadays, the World Wide Web is the main source for people to access information. Certainly, search engines like Yahoo![1] and Google[2] belong to the pioneers of web search technology by creating retrieval systems, which enable users to gather information from the web in a few milliseconds. With the increasing popularity of the Web 2.0 applications such as social networks, blogs, and wikis, Web users have today the possibility not only to consume, but also to share information.

The rapidly growing amount of information on the web brings new challenges for search engines and social networks to make information accessible, manageable, and even enjoyable for the user. Since the success of web applications highly depends on whether their services meet the user's expectations, research and industry invest significant efforts to better understand the user's motivations, intentions, and behavior for improving the quality of their methods and algorithms.

Understanding the users' search intention plays a crucial role in retrieval systems in order to identify relevant information for the user. In a typical search scenario, a query is submitted by the user to a retrieval system, which in turn returns a ranked set of candidate answers [CMS09] based on their relevance to the user query. In modern search engines, users expect to access information very fast. Therefore, search engine architectures have to consider both, the quality of results (effectiveness) and the speed of answering a search request (efficiency) [CMS09]. The search query is used for identifying relevant information from a large collection of information. Since the query is often very short, search engines use additional information, e.g. by extracting features from the documents (e.g. web sites) and other available meta-data, to identify relevant information. For example in learning-to-rank, hundreds of features are used to learn a model for ranking the documents for a given query [CC11]. In this context, one of the challenges is to identify a subset of relevant features, which can improve the quality of the final ranking, and simultaneously reduce the computational time.

[1] https://www.search.yahoo.com

[2] https://www.google.com

Another issue for retrieval systems is that the search query is not always reflecting the "real" search intention of the user, e.g. the case of ambiguous queries. For example a search query "python" can lead to a set of search results related to "python animal", while the user may be looking for content related to "python programming" [SMO10a, SMO11]. Even for a query "python programming", users can have very diverse intents, e.g. looking for introductory tutorial or looking for tutors, books, etc.[SMO10a]. This example shows that queries often can be interpreted wrong from a retrieval system. Both challenges related to feature selection for ranking and diversification of web search results are critical issues for search engines that have to reduce the risk that users' search request stays unsatisfied.

Web 2.0 applications such as Facebook[3], Twitter[4], and LinkedIn[5] allow the user not only to consume, but also to share personal and business related information. Furthermore, social networks allow people to build subgroups or communities based on different professions, topics, resources and cultural similarity. Similar to traditional online websites (e.g news), social networks have to keep the users active on their platform (e.g. for watching advertisements [CGGG17])to succeed in the market. Therefore, social networks try to better understand the users' motivation for using their platforms to better support the user in their needs and expectations. However, users have different motivations for using a particular social network platform [SO13], and this along with the rapidly increasing amount of information brings new challenges for online social networks to support the users in their daily and long-term usage of their services.

Privacy is another issue in social networks in which people share a significant amount of personal content, e.g. messages, photos, videos, etc. Studies on users' privacy management show that people often ignore the privacy settings of their content before sharing it, even though studies show that they are concerned about their privacy [ZSN$^+$13, LGKM11, Mad12]. Social networks often offer the user to set different types of privacy settings before sharing it. However, studies on privacy settings management show that these functionalities are often too time consuming or confusing for the user [MJB12].

Protecting the privacy of the user is one of the challenges in social networks. Studies on users' privacy management show that in a daily use of social networks people are ignoring the privacy of their content shared in the network [ZSN$^+$13], which brings the risk of sharing highly private information into a wider audience, e.g. through users carelessness [LGKM11, Mad12]. The management of the shared information using the privacy setting is often time costly and confusing for the user [MJB12]. Further, the users have different opinion regarding what kind of content should be considered private. This motivates to build applications that can support the user in their privacy decisions.

[3] https://www.facebook.com/

[4] https://twitter.com

[5] https://www.linkedin.com

Another challenge for users in social networks is to keep track of their personal information over a long period of time. From a long term perspective, the social networks contain a personal archive of the user including different facets of life [ZSN+13]. Recent studies show that popular social networks such as Facebook and Twitter refresh their trending topics every 10 to 15 minutes, which leads to a lack of coverage in information presented to the user [CGGG17]. This information flood has the effect that information management in social networks is very challenging for the user, since content such as messages, news, and photos get only a short time attention and then become forgotten in the future.

1.1 Outline of the Thesis

In this thesis, we contribute to core challenges of modern web applications for accessing, sharing, and managing information. The rest of the thesis is organized as follows. Chapter 2 gives an overview of the approaches and state-of-the-art techniques used in this thesis, including brief introduction to information retrieval and machine learning.

In Chapter 3, we present an in-depth study of web search results diversification with the focus of scalability and efficiency in large-scale web search engines. First, we propose a clustering based approach to reduce the computational time of an implicit diversification algorithm to achieve linear complexity. Second, we investigate the problem of web search diversification methods in a distributed setup for large-scale IR environments. The work reported in Chapter 3 is published in:

- [NAS16] Kaweh Djafari Naini, Ismail Sengor Altingovde, and Wolf Siberski. Scalable and efficient web search result diversification. *ACM Transactions on the Web*, *TWEB*, 10(3):15:1-15:30, August 2016.

In Chapter 4, we contribute to the problem of feature selection in learning-to-rank to improve effectiveness and efficiency. The approach presented in this thesis is based on a set of different diversification algorithm. The assumption is that diversification of features can improve the quality of the learning-to-rank models. The work reported in Chapter 4 is published in:

- [NA14] Kaweh Djafari Naini and Ismail Sengor Altingovde. Exploiting result diversification methods for feature selection in learning to rank. In *Proceedings of the 36th European Conference on Information Retrieval*, *ECIR'14*, pages 455-461, 2014.

In Chapter 5, we address the problem of privacy protection in social network. In this context, we envision an application that can suggest the user the right privacy setting. To overcome this issue, we present a thought analysis of privacy settings in social web. Further, we present an approach to predict the privacy setting of the content before it is shared by the user. The work reported in Chapter 5 is published in:

- [NAK$^+$15] Kaweh Djafari Naini, Ismail Sengor Altingovde, Ricardo Kawase, Eelco Herder, and Claudia Niederée. Analyzing and Predicting Privacy Settings in the Social Web. In *Proceedings of the 23rd International Conference on User Modeling, Adaptation and Personalization, UMAP'15*, pages 104-117, Dublin, Ireland, June 29 - July 3, 2015.

In Chapter 6, we present an in-depth analysis about the communications practices in social and professional networks. The focus of the analysis is on commonalities and differences between different networks. This work is relevant for interpreting results from social media for identifying group-specific resources. The work reported in Chapter 6 is published in:

- [CHNS14] Sergiu Chelaru, Eelco Herder, Kaweh Djafari Naini, and Patrick Siehndel. Recognizing skill networks and their specific communication and connection practices. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media, HT'14*, pages 13-23, Santiago, Chile, 2014.

In Chapter 7, we discuss the challenge of identifying content in social media for generating life summaries. In this work, we analyze a corpus of social media posts to identify a set of features which characterize memorable posts. Next, we apply general and personalized machine-learning models for ranking posts for retention. The work reported in Chapter 7 is published in:

- [NKK$^+$18] Kaweh Djafari Naini, Ricardo Kawase, Nattiya Kanhabua, Claudia Niederée, and Ismail Sengor Altingovde. Those were the days: learning to rank social media posts for reminiscence. *Information Retrieval Journal*, pages 1-29, 2018.

- [NKKN14] Kaweh Djafari Naini, Ricardo Kawase, Nattiya Kanhabua, and Claudia Niederée. Characterizing high-impact features for content retention in social web applications. In *23rd International World Wide Web Conference, WWW'14*, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume, pages 559-560, 2014.

- [NKTN18] Claudia Niederée, Nattiya Kanhabua, Tuan Tran, and Kaweh Djafari Naini. Preservation Value and Managed Forgetting. In *Book of Personal Multimedia Preservation*, 2018, pages 101-129.

Throughout the course of my PhD I also have contributed to other publications related to information retrieval, data mining and social network analysis:

- Asmelash Teka Hadgu, Kaweh Djafari Naini, and Claudia Niederée. Welcome or not-welcome: Reactions to refugee situation on social media. *CoRR*, arXiv, abs/1610.02358, 2016.

- Ujwal Gadiraju, Kaweh Djafari Naini, Andrea Ceroni, Mihai Georgescu, Dang Duc Pham, and Marco Fisichella. Wikipevent: Temporal event data for the semantic web. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference*, *ISWC 2014*, Riva del Garda, Italy, October 21, 2014, pages 125-128.

- Andrea Ceroni, Mihai Georgescu, Ujwal Gadiraju, Kaweh Djafari Naini, and Marco Fisichella. Information evolution in wikipedia. In *Proceedings of The International Symposium on Open Collaboration*, *OpenSym 2014*, Berlin, Germany, August 27 - 29, 2014, pages 24:1-24:10.

- Tuan A. Tran, Andrea Ceroni, Mihai Georgescu, Kaweh Djafari Naini, and Marco Fisichella. Wikipevent: Leveraging wikipedia edit history for event detection. In *Proceedings of the 15th International Conference Web Information Systems Engineering*, *WISE 2014*, Thessaloniki, Greece, October 12-14, 2014, Proceedings, Part II, pages 90-108.

- Ernesto Diaz-Aviles, Patrick Siehndel, and Kaweh Djafari Naini. Exploiting social #-tagging behavior in twitter for information filtering and recommendation. In *Proceedings of The Twentieth Text REtrieval Conference*, *TREC 2011*, Gaithersburg, Maryland, USA, November 15-18, 2011.

2

Technical Background

In this chapter, we introduce the technical background of the research conducted in this thesis. The chapter is structured in two main parts. First in Section 2.1, we give an overview of approaches in information retrieval and web search engines. Second in Section 2.4, we describe the concept of supervised and unsupervised machine learning.

2.1 Information Retrieval (IR) and Web Search

The field of information retrieval deals with the problem of finding relevant content from within large collections of unstructured and/or semi-structured data that satisfies the user's information need [MRS08]. The most common information retrieval application is Web search that allows the user to retrieve documents from the web using a search query [CMS09]. Towards the fact that search engines apply many of IR techniques to improve the effectiveness or quality of their search system, they have to perform efficient to be able to answer a search request as fast as possible [CMS09]. In the figure 2.1 we describe the issues of information retrieval and search engines design according to Croft et al. [CMS09].

In the following sections, we describe some of the most popular IR models e.g. vector space model and BM25. Further, we briefly describe the core tasks of a modern search engines. It has to be noted that we are not describing all the issues presented in the figure 2.1 since this would be out of the scope of this work.

2.1.1 Vector Space Model

The vector space model first proposed by [SWY75] is the concept of representing documents and queries in a n-dimensional vector space with n representing the number of unique terms in the entire collection [CMS09]. The corresponding vectors for each documents contain values corresponding for each term, e.g. *term-frequency*

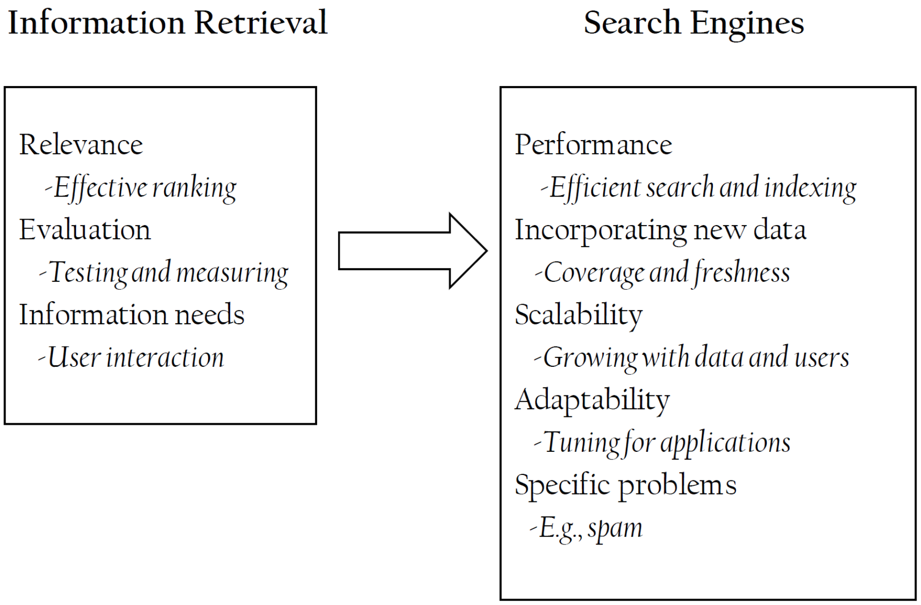


Figure 2.1. Core issues for information retrieval and search engine design [CMS09].

or *tf-idf* which is term-frequency normalized by the *inverse document frequency* (idf) [CMS09, MRS08].

The tf-idf measure for each term t and each document d is defined as follows:

$$tf\text{-}idf = tf.\log\frac{N}{df_t}, \tag{2.1}$$

where tf is the term frequency, df the document frequency of the term t, and N being the size of the collection [MRS08].

The vector space model allows to compute different similarity and distance measures between two documents or between a document and a query with the same t-dimensional representation. Given a document d and a query q we define the cosine similarity of two vectors $\vec{q}$ and $\vec{d}$ of the same length as follows:

$$\text{cosine-sim}(\vec{q},\vec{d}) = \frac{\vec{q}.\vec{d}}{|\vec{q}|.|\vec{d}|}, \tag{2.2}$$

where the numerator is the dot product of the vectors and the denominator is the product of their euclidean lengths [MRS08]. By using tf-idf vectors the cosine similarity value is in the range of 0 to 1 [MRS08].

Olympia	1:1	2:3		
Rio	1:1	2:1	3:1	4:1

Table 2.1. Examples for the inverted index of words.

2.1.2 Indexing

The indexing includes several core components for example crawling, transforming, index creation, index inversion, and index distribution [CMS09]. Here we describe the creation, inversion, and distribution of the index. The creation of the index can be done using the weights described in the context of vector space model in Section 2.1.1. Modern search engines extract the document vectors using the inverted index strategy. The inverted index stores each term t in a *posting list* pointing to the documents containing the term t [CMS09, MRS08]. The inversion component creates the inverted index by transforming the stream of document-term information into a term-document information. The index distribution component uses a distributed system including multiple servers across multiple sites of a network [CMS09]. A distributed architecture is unavoidable for modern search engines for efficient performance and parallel processing of the queries [CMS09].

In table 2.1 we present an example for the inverted indexes of the words *Olympia* and *Rio*. The term *Olympia* appears once in the document d_1 and three times in the document d_2, whereas the term *Rio* appears in all the four documents once.

2.1.3 Top-K Retrieval and Ranking

The ranking algorithms in IR aim to retrieve and rank a set of documents for a given search query. For instance, a simple ranking function can use tf-*idf* scores for sorting the documents based on their tf-*idf* score for the query, starting with the highest score on top. Another popular ranking function is Okapi BM25 [RWJ+95] which has performed well in TREC retrieval experiments [1] and has influenced the ranking algorithms of commercial search engines, including web search engines [CMS09].

The BM25 scoring function is defined as follows:

$$\sum_{i \in Q} log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} . \frac{(k_1 + 1)f_i}{K + f_i} . \frac{(k_2 + 1)qf_i}{k_2 + qf_i}, \tag{2.3}$$

where the summation is over all terms appearing in the query [CMS09]. The parameters are described in the following list.

[1] http://trec.nist.gov/

i:	term,
q:	query,
R:	the number of relevant documents for this query,
r_i:	the number of relevant documents containing term i,
N:	the total number of documents in the collection,
n_i:	the number of documents containing term i,
k_1,k_2,K:	parameters set empirically,
f_i:	the frequency of term i in the document,
qf_i:	the frequency of term i in the query q,

In Section 2.4, we describe another approach for improving ranking using a supervised machine learning approach.

2.2 Diversification

In Chapter 3 and Chapter 4, we present approaches for improving retrieval effectiveness and efficiency using diversification algorithms. In this section, we introduce the problem of web search result diversification and present different types of diversification algorithms.

2.3 Diversification of Web Search Results

Traditional IR systems often try to rank the documents by maximizing the relevance for a given search query [CG98b]. Considering a retrieval scenario with only a few relevant documents, or a system requiring high recall, relevance is usually a good indicator for retrieving information [CG98b]. However, in search engines where the user only uses a few query terms on a large collection of web documents, using the relevance alone comes with the risk that by a wrong interpretation of the query, the system cannot satisfies the user's information need [GS09]. Recent studies on Web search results diversification aim to minimize this risk by creating a set of relevance but diverse set of search results [SMO10a, AGHI09, CKC+08b].

The problem of *search result diversification* can often be described by a *trade-off* between the relevance of the documents to the query and the diversity between the documents within the result set [VRB+11]. A general definition of the problem of Web search results diversification given by Santos et al. [SMO15, San13] is as follows:

Given a set of ranked documents $D = d_1 ... d_n$ with n elements for a query q by a relevance orientated approach, and given N_q and N_d the set of information needs for which the query q and each $d \in D$ are relevant. The diversification aim to find a subset $S \in D$ such that:

ALGORITHM 1: MMR: The Maximal Marginal Relevance Diversification as Presented in [VRB+11]

```
Input  : $D, k$
Output: $S$
$S \Leftarrow \emptyset$
$d' \Leftarrow \text{argmax}_{d_i \in D}\, mmr(d_i)$
$D \Leftarrow D \backslash d'$
$S \Leftarrow d'$
repeat
    $d' \Leftarrow \text{argmax}_{d_i \in D}\, mmr(d_i)$
    $D \Leftarrow D \backslash d'$
    $S \Leftarrow S \cup d'$
until $|S| < k$
return S
```

$$S = \underset{S' \in 2^D}{\text{argmax}} \left| \bigcup_{d \in S'} N_q \cap N_d \right|, s.t. |S'| \leq k, \tag{2.4}$$

where $k > 0$ is the *diversification cutoff*, D the number of top initial ranked documents, and 2^D is the power set of D containing all subsets (candidate permutations) S' of D, with $0 < |S'| \leq k$. The optimal diversified set S is the set of maximum number of covered information up to the cutoff k.

Complexity Analysis. The diversification problem is an instance of the maximum coverage problem which is NP-hard in computational complexity theory [Hoc97, San13]. Agrawal et. al. [AGHI09] show that the diversification problem can be reduced to the problem of Max Coverage which is a well known NP-hard problem. To overcome this problem there are several proposed studies on diversification using for example *best first-search* approaches such as *Maximal Marginal Relevance* MMR [CG98b].

In general, there are two main types of diversification algorithms, *implicit* and *explicit* [SMO10a]. Implicit diversification assumes that similar documents cover similar aspects, and should be denoted in the final ranking by reducing the overall redundancy [SMO10a], while explicit diversification uses different query aspects to maximize the coverage in the final result set with respect to these aspects [SMO10a].

In the following, we present three diversification methods, MMR [CG98a] and MSD [GS09] for implicit diversification and xQuAD for explicit diversification [SMO10a].

Maximal Marginal Relevance (MMR). The MMR diversification proposed by Carbonell et al. [CG98a] is presented in Algorithm 1 and defined as follows. Given an initial set of Documents D for the query q, the MMR algorithm first select the most relevant document and add it to the output set S. Then the algorithm loops

ALGORITHM 2: MSD: The Max-Sum Dispersion (MSD) as Presented in [VRB+11]

```
Input  : $D, k$
Output: $S \subseteq D, |S| = k$
$S \Leftarrow \emptyset$
repeat
    $\{d_i, d_j\} \Leftarrow \mathrm{argmax}_{d_i, d_j \in D}\, msd(d_i, d_j)$
    $S \Leftarrow S \cup \{d_i, d_j\}$
    $D \Leftarrow D \setminus \{d_i, d_j\}$
until $|S| < \lfloor k/2 \rfloor$
if $k$ is odd then
    choose an arbitrary object $d_i \in D$
    $S \Leftarrow S \cup d_i$
end
return $S$
```

over all the candidate documents and select a document which maximize the following function:

$$mmr(d_j) = (1 - \lambda) rel(d_j) + \frac{\lambda}{|D|} \sum_{d_i \in D} \delta_{div}(d_i, d_j), \tag{2.5}$$

with $\lambda \in [0, 1]$ as the trade-off parameter, $rel(d_j)$ the relevance of the document d_j for the query q, and $\delta_{div}(d_i, d_j)$ the diversity function. The diversity function $\delta_{div}(d_i, d_j)$ measures the diversity between two documents, often defined by the distance function $\delta_{div}(d_i, d_j) = 1 - sim(d_i, d_j)$. The $sim(d_i, d_j)$ can be for example the cosine similarity as described in the equation 2.2. One issue with the MMR approach is that the first document with the highest relevance is always included in the result set which has high influence of the followed selection of the documents [VRB+11].

Max-Sum Dispersion (MSD). The greedy approach MSD diversification proposed by [GS09] is presented in Algorithm 2. The MSD algorithm assigns in each around a pair of documents into the result set, which are relevant to the query and diverse to each other. Incrementally in each iteration two candidate documents are selected that maximize the following equation:

$$msd(d_i, d_j) = (1 - \lambda)(rel(d_i) + rel(d_j)) + 2 * \lambda \delta_{div}(d_i, d_j) \tag{2.6}$$

In the case the number of the documents is odd. Therefore, in the final step the algorithm includes an arbitrary element into the result set R.

Other diversification approaches show that *greedy local search* can outperform the best-first search approaches mentioned above [ZAZW12]. In Chapter 3, we discuss diversification algorithm implemented as greedy local search more in detail.

ALGORITHM 3: xQuAD: The Explicit Query Aspect Diversification [SMO10a]

```
Input : $q, D, k, \lambda$
$S \Leftarrow \emptyset$
repeat
    $d' \Leftarrow \text{argmax}_{d \in D \setminus S}\,(1-\lambda)P(d|q) + \lambda P(d,\bar{S}|q)$
    $D \Leftarrow D \backslash d'$
    $S \Leftarrow S \cup d'$
until $|S| < k$
return $S$
```

As mentioned before, explicit result diversification uses different query aspects as information to build a relevant but diverse set of documents for a given query. The *Explicit Query Aspect Diversification* (xQuAD) is probabilistic framework for explicit result diversification proposed by Santos et al. [SMO10a].

Explicit Query Aspect Diversification (xQuAD). The xQuAD algorithm proposed by Santos [SMO10a] is presented in Algorithm 3 and described in the following equations 2.7–2.11 [SMO10a]. For a given query q and set of initial ranking D, the algorithm creates a new set of ranked documents with the size limited by the variable k which maximize the equation 2.7. The parameter λ is used to control the trade-off between relevance and diversity.

$$(1-\lambda)P(d|q) + \lambda P(d,\bar{S}|q) \tag{2.7}$$

The function $P(d,\bar{S}|q)$ measures the relative importance of a sub-query q_i from the set of all the sub-queries of the query q. As mentioned before explicit diversification uses not only the relevance, but takes into account additional aspects from the query in this case the set of sub-queries $Q = \{q_1 \ldots q_n\}$.

$$P(d,\bar{S}|q) = \sum_{q_i \in Q} [P(q_i|q)P(d|q_i) \prod_{d_j \in S} (1 - P(d_j|q_i))] \tag{2.8}$$

with $P(q_i|q)$ measuring the relative importance of the sub-query q_i compared to the other sub-queries in the set and $P(d,\bar{S}|q)$ provides a probability for each document which is not already being selected in S.

$$P(d,\bar{S}|q_i) = P(d|q_i)P(\bar{S}|q_i) \tag{2.9}$$

The computation of $P(\bar{S}|q_i)$ is presented in the following equation:

$$P(\bar{S}|q_i) = \prod_{d_j \in S} (1 - P(d_j|q_i)) \tag{2.10}$$

The probability for a sub-query over all the documents is independent of the relevance of other documents in S to the same sub-query.

$$(1-\lambda)P(d|q) + \lambda \sum_{q_i \in Q} [P(q_i|q)P(d|q_i) \prod_{d_j \in S} (1 - P(d_j|q_i))] \tag{2.11}$$

There are several measures for evaluating the quality of diversification such as *α-Normalized Discounted cumulative gain* (α-NDCG) [CKC^{+}08a], *S-recall* [CMZG09a], and *Expected Reciprocal Rank-IA* (ERR-IA) [ZCL03].

2.4 Machine Learning (ML)

In this section, we briefly describe the concept of machine learning and present the set of machine learning methods applied in this thesis. The general concept of *learning* is defined as follows:

A computer program is said to *learn* from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E [Mit97].

Machine learning approaches try to avoid bias and noise effects in their models, e.g. using various of statistical approaches. This problem is called *overfitting* and defined as follows:

Given a hypothesis space H, a hypothesis $h \in H$ is said to overfit the training data if there exists some alternative hypothesis $h^{'} \in H$, such that h has smaller error than $h^{'}$ over the training examples, but h has a smaller error than h over the entire distribution of instances [Mit97].

There are two types of machine learning algorithms based on supervised and unsupervised learning. In supervised learning the input of the method is a set of labeled (annotated) data called *training* set. The labels in the training set present the actual outcome for a subset of the data. For example in text *classification*, a training set can be created by manually annotating documents as “relevant” or “not-relevant” for a subset of data. In this case, the algorithm use the labeled data to *learn* a pattern for identifying “relevant” and “not relevant” documents. The identified pattern is then a *model* which can be applied on new data, e.g documents, to classify them. The quality of a generated model is usually validated on a holdout set called *test* set. The test set is an annotated set of the data which is not used in the model generation.

Popular examples for unsupervised methods are clustering algorithms that group a set of items (e.g. documents) into subsets or *clusters*. A standard document clustering task is to build clusters that are internally coherent but clearly different from each other [MRS08].

2.4.1 Classification

In this section, we present three machine learning algorithms and Naive Bayes and REP(Tree) for classification

Naïve Bayes Classifier. Naive Bayes classifier is one of the most popular methods in the area of Bayesian learning. The main idea of Bayesian learning is based on the Bayesian theorem for calculating the probability of an event based on conditions observed in the training data [Mit97]. The naïve Bayes classifier is "naïve" since it assumes that the attribute values are conditionally independent from the targeted class [WFH11]. For example for an given targeted value v_j with the attributes $a_1 \dots a_n$ the probability of observing the targeted class is the product of probabilities of the attributes: $P(a_1, a_2 \dots, a_n | v_j) = \prod_i P(a_i | v_j)$. Using the Bayesian theorem we have the following equation for the Naive Bayes Classifier:

$$v_{NB} = \operatorname*{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \tag{2.12}$$

where v_{NB} describes the targeted value of the naive Bayes classifier [Mit97]. By a given instance from the test set we can estimated the targeted class by using the features of the new instance to calculate the probability of belonging to one or another class [Mit97].

REPTree. The decision tree learning is characterized by the fact that the learning model is represented by a decision tree [Mit97]. The nodes in the decision trees are the attributes. A popular decision tree algorithm is *ID3* which is using information gain to identify which attribute should be selected at each stage while growing the tree [Mit97]. However, the ID3 strategy might led to some problems once the training data has some noise or the size of the data is too small to have representative sample for a targeted class [Mit97]. Therefore new decision tree algorithms try to prevent overfitting by using the pruning approaches such as *Reduced-error pruning* which is mainly removing nodes and subtrees which might include "noise" from the data [Mit97]. Pruning of the tree is also reducing the complexity of the tree which is leads to more efficient performance. In this thesis, we apply the REPTree algorithm which is similar to ID3 using the information gain to build the model by using reduce error pruning to avoid noise [WFH11].

2.4.2 Learning-To-Rank (LETOR)

In this section, we present a machine learning approach for ranking using Support Vector Machines (SVM).

As discussed in the previous section, ranking is one of the most important tasks in information retrieval systems. In the previous section 2.1, we introduced the concept of ranking by using relevance scores and unsupervised ranking functions such as

BM25. These ranking approaches provide a reasonable ranking of documents over a large collection of data using only a few features. This makes unsupervised methods very efficient and useful to large-scale retrieval systems for retrieving a first candidate set of documents before applying supervised machine learning methods (also known as two-stage ranking [DBC13]). The study of ranking using supervised machine learned models is called *learning-to-rank* and often shortened to *LETOR*. In learning-to-rank all the documents are represented as feature vectors. Further, the training set includes query, document, relevance judgment, and a set of features. The training set contains a set of n training queries $\{q_i\}_{i=1}^{n}$, with the feature vector representation of the documents to a query, $x^{(i)} = \left\{x_j^{(i)}\right\}_{j=1}^{m^{(i)}}$, where $m^{(i)}$ is the number of documents belonging to q_i the corresponding relevance judgment [Liu09]. In [Liu09], the authors distinguish between three categories of LETOR logarithms, Pointwise (e.g. gradient boost [PL08]), Pairwise (e.g LambdaRank [BRL07]), and Listwise (e.g AdaRank [XL07]). In this thesis, we use a popular pairwise algorithm RankSVM based on the *Support Vector Machine* (SVM [Joa06]) which is often used in web search applications with large number of training instances and features [CMS09].

Ranking Using Support Vector Machine (SVM). Joachim introduces a pairwise approach for learning-to-rank using support vector machine [Joa02a], presented in the following equations 2.13–2.14. The training set of size n contains a set of queries q with their target rankings r^* $(q_1, r_1), (q_2, r_2), \ldots, (q_n, r_n)$.

For each pair of documents, we define $(d_i, d_j) \in r_i$ if d_i has an higher rank than d_j otherwise $(d_i, d_j) \notin r_i$. Now we would like to find a vector $\vec{w}$ that satisfies many of the following conditions as possible:

$$\begin{aligned} \forall (d_i, d_j) \in r_1 : \vec{w}.\phi(q, \vec{d_i}) &> \vec{w}.\phi(q, \vec{d_j}) \\ &\cdots \\ \forall (d_i, d_j) \in r_n : \vec{w}.\phi(q, \vec{d_i}) &> \vec{w}.\phi(q, \vec{d_j}) \end{aligned} \tag{2.13}$$

$\phi(q, \vec{d_i})$ describes the match between query and document d.

It has shown the above problem is NP-Hard [HSV95]. However, this problem can be formalize it as a SVM optimization problem:

$$\begin{gathered} minimize : \frac{1}{2}\vec{w}.\vec{w} + C \sum \xi_{i,j,k} \\ subject\ to : \\ \forall (d_i, d_j) \in r_1 : \vec{w}.\phi(q, \vec{d_i}) > \vec{w}.\phi(q, \vec{d_j}) + 1 - \xi_{i,j,1} \\ \cdots \\ \forall (d_i, d_j) \in r_n : \vec{w}.\phi(q, \vec{d_j}) > \vec{w}.\phi(q, \vec{d_j}) + 1 - \xi_{i,j,n} \\ \forall_i \forall_j \forall_k : \xi_{i,j,k} \geq 0 \end{gathered} \tag{2.14}$$

where ξ is a *slack variable* which allows miss-classification in the training examples, and C is for allowing trading-off margin size for avoiding overfitting.

There are several measures for evaluating the quality of ranking models such as *Normalized Discounted cumulative gain* (NDCG) [JK02a], *Mean Average Precision* (MAP) [BYRN99], and *Expected Reciprocal Rank* (ERR) [CMZG09b].

2.4.3 Feature Selection for LETOR

Feature selection is an important aspect in machine learning and falls into three different categories, namely, filter, wrapper and embedded approaches [GLQL07]. In contrast to the wrapper and embedded approach, filter-based feature selection considers features independently from the classifiers in a preprocessing step. Earlier work on feature selection show feature selection can improve efficiency and accuracy of classifiers, and led to diversity in ensemble learning [CC00]. In this thesis, we focus only on feature selection methods fall into the filter category for learning-to-rank.

In this section, we introduce a greedy strategy proposed by Geng et al. in [GLQL07] defined as follows. For each feature a relevance score is computed using a measure such as NDCG [JK02a]. For comparing the similarity between two features $sim(f_i, f_j)$, we can use the the Kendall's Tau [KEN38] distance between their top-k rankings averaged over all queries as described in [GLQL07]. Given a set of features, the GAS algorithm first select the feature with highest average relevance score into the set F_k. Next, for each of the remaining features f_j, the relevance score is updated with respect to the following equation:

$$rel(f_j) = rel(f_j) - sim(f_i, f_j) \cdot 2c, \tag{2.15}$$

where c is a parameter to balance the relevance and diversity optimization objectives. The GAS algorithm is a greedy algorithm and stops when k features are selected.

2.4.4 Clustering

k-Means. The k-Means algorithm (also called Lloyds algorithm) is one of the most popular clustering methods [HW79] in information retrieval. k-Means assigns a set of documents $d_i, \ldots, d_n$ to k number of clusters. In the first step, k-Means is randomly selecting k documents as initial seeds of the clusters. Then, k-Means computes the distance between each document and centroid to find the cluster which has the closest distance to the document. Each document is then assigned to its closest cluster. In the final step, k-Means calculate the new centroids by taking the mean of documents assigned to the cluster. The last two steps are then repeated until some stopping criteria is met.

The distance function usually defined as the Euclidean distance between two the document d_i and the centroid Cluster C as follows:

$$dist(d_i, C) = ||d_i - ctr(C)||^2 \tag{2.16}$$

with ctr(C) defined as the centroid of the cluster C.

The centroid of a cluster is defined as the mean of the documents in the cluster:

$$ctr(C) = \frac{1}{|C|} \sum_{d \in C} \vec{d}. \tag{2.17}$$

with $|C|$ defined as the number of the documents in the cluster C [MRS08].

The complexity of *k-Means* is defined by the number of iteration I, number of clusters K, number of vectors N that gives a linear complexity of $O(KNI)$ [MRS08]. In document clustering, the algorithm converges often very fast since the document vectors are sparse which makes the distance computation very fast [MRS08]. However, in some cases the distance computation can be even faster by taking a document closest to the centroid as the new centroid as it is done by the k-Medoids algorithm [MRS08].

k-Nearest Neighbor (kNN). The k-Nearest Neighbor clustering algorithm is an unsupervised ML method. The algorithm is building for each input element, a cluster with k closest neighbor elements[CMS09]. In contrast to the k-Means clustering, the kNN algorithm can produce overlapping clusters. In chapter 7, we apply kNN for identifying users' nearest neighbors for personalized ranking.

3

Scalable and Efficient Web Search Result Diversification

In Chapter 2.2, we introduced the problem of web search result diversification and presented approaches for improving the ranking effectiveness by taking not only relevance, but diversity into account. However, currently proposed diversification approaches have not put much attention on practical usability in large-scale systems such as modern search engines. In this Chapter, we present two contributions towards this goal. First, we propose a combination of optimizations and heuristics for an implicit diversification algorithm based on the desirable facility placement principle, and present two algorithms that achieve linear complexity without compromising the retrieval effectiveness. Second, we describe and analyze two variants for distributed diversification in a computing cluster, for large-scale IR where the document collection is too large to keep in one node.

3.1 Introduction

The success of a search engine in a highly competitive market is tightly bound to how effectively its top ranked answers satisfy the user's information need. Not surprisingly, considerable research effort is devoted to ranking candidate answers and determining the optimal top-k results both by the academia and industrial players. A recent yet well-recognized aspect in this sense is diversifying the top search results, especially when the user's search intent is not clear, which has its roots in minimizing the risk in a financial portfolio [Mar52]. That is, just like an investor who is not sure about the future diversifies the selection of stocks in her portfolio, a search system that cannot predict the search intent behind a query should diversify the top search results to minimize the risk of frustrating its users [WZ09a].

While the classical examples for such ambiguous queries include *java* (or *jaguar*, or *apple*), where a search system should return answers related to *Java programming* *language* and *Java island*, it is soon realized that diversity is needed at different levels even for queries that look much less ambiguous at the first glance [SMO10a, SMO11]. For instance, users submitting the query *java programming* can still have very diverse intents, such as finding an introductory tutorial, obtaining pointers to some resources like books or class-notes, discovering forums, checking ads for tutors, and so on [SMO10a].

The above example demonstrates that most keyword queries would inherently involve some ambiguity, to a lesser or greater extent, and hence can benefit from the promises of result diversification. As this carries diversification from a niche-operation to a widely used everyday task for large-scale search engines, the need for efficient and scalable algorithms becomes inevitable. Advances are required in two areas: first, the computational complexity of diversity algorithms needs to be reduced to fit in the tight budget of online query processing (usually a few hundred milliseconds), and second, these algorithms need to be adapted to the computing cluster architecture established for search engines.

The contributions in this chapter are thus two-fold: first, we improve the efficiency of a state-of-the-art implicit result diversification algorithm based on the desirable facility placement principle (from Operations Research) solved by a Greedy Local Search (GLS) heuristic [ZAZW12]. Recently, this algorithm has been shown to have an impressive effectiveness for identifying relevant and novel top-k results, but its quadratic cost with the number of candidate documents renders this algorithm impractical for real-world usage. We propose simple yet effective optimizations that employ pre-clustering of the candidate documents for improved efficiency (i.e., linear with the number of candidate documents) without sacrificing the effectiveness. In a practical setting where top-10 (or 20) results are selected from a candidate set of a few hundred (or thousand) documents, our optimized algorithms, so-called C-GLS and C^2-GLS, can reduce the online query diversification cost by more than 80%, and for some cases, up to 97%.

As a second contribution, we turn our attention to incorporating the diversification algorithms into a large-scale search system that would typically operate on a cluster of thousands of machines. While diversification algorithms in the literature are extensively evaluated in terms of their effectiveness, the impact of the distributed architecture on which they need to operate has not been addressed yet. In contrast, the effectiveness and efficiency of the diversification algorithms may also depend on the architecture and more specifically, the layer where the actual diversification is realized. We introduce two possible strategies, broker-based and node-based diversification, and identify potential effectiveness and efficiency trade-offs for both implicit and explicit diversification algorithms. To be comparable with the previous studies in the literature, our strategies are evaluated using the standard experimental framework employed in the TREC Diversity Tasks in 2009 and 2010. To the best of our knowledge, our contribution in this direction is pioneering, as there exists no earlier work in the literature that investigates the diversification performance on top of a distributed architecture.

The rest of the Chapter is organized as follows. In Section 3.2, we first provide a review of search result diversification algorithms and then outline the principles of query processing techniques in widely employed distributed search systems. Next, we briefly summarize the GLS based algorithm from [ZAZW12] and then introduce our more efficient variants in Section 3.3. Section 3.4 is devoted to two distributed diversification strategies, namely, broker-based and node-based diversification. In Section 3.5, we experimentally evaluate proposed strategies. Finally, we summarize our key findings in Section 3.6, and conclude and point out future research directions in Section 3.7.

3.2 Related Work

Approaches for search result diversification can be categorized as either *implicit* or *explicit* [SMO10a]. Given a set of candidate documents retrieved for a query, implicit methods aim to discover the possible different aspects from these documents in an unsupervised manner. In contrast, explicit diversification methods directly model the query aspects, exploiting external knowledge such as manually or automatically assigned query labels or query reformulations. In Sections 3.2.1 and 3.2.2, we review algorithms for implicit and explicit search result diversification, respectively. In Section 3.2.3, we provide a brief outline of distributed search on computing clusters, on which such diversification algorithms are intended to operate.

3.2.1 Implicit Result Diversification Techniques

The basic assumption of earlier IR ranking, also underlying the probability ranking principle (PRP, [Rob77]) assumes that the probability of a document being relevant is independent from the other documents in the result set. However, when viewing the information retrieval task as a decision process where the task is to minimize the risk that a user's information need remains unsatisfied, it becomes clear that a result set with high coverage entails a lower risk than a result set with the most relevant, but also very similar documents. Zhai and Lafferty [ZL06] present a formal model for this risk minimization approach, where they introduce the notion of loss as the degree to which a search effort fails to satisfy a user's information needs. In this model, different diversification objectives become loss functions which probabilistically estimate the loss incurred by a user for a given result set.

Several such objectives are proposed and successfully validated. The use of Maximum Marginal Relevance (MMR) is proposed by Carbonell and Goldstein [CG98b], and also used in Zhai et al. [ZCL03]. Chen and Karger [CK06] maximize the probability of retrieving at least one relevant document by assuming that all documents that are ranked higher than the current one as irrelevant (i.e., as a form of negative feedback). Carterette and Chandar [CC09] focus on a finer grain level of diversification, similar to the examples given in Section 3.1, and introduce the so-called faceted topic retrieval. In this case, the query facets are identified using LDA and relevance modeling, and then PRP is employed while assessing the coverage of facets by the candidate documents. In a similar fashion, He et al. [HMdR11] first cluster the candidate documents and then select the best clusters to apply diversification to improve the final result quality. Carpineto et al. [CDR12] compare some well-known implicit diversification techniques (e.g., based on MMR) to those that cluster the candidate documents and select the cluster representatives into the final result list. Besides other metric space based methods, Gil-Costa et al. [GCSMO11, GCSMO13] describe a diversification approach that again clusters the candidate documents and constructs the diversified result where the cluster centroids are placed at the top of the list. Different from all the latter works that essentially employ greedy *best-first search* diversification methods (such as the round-robin strategy or MMR) on top of the document clusters, our work presented here exploits clustering as a preprocessing stage to improve the efficiency and scalability of a greedy *local search* method, which is shown to outperform the best-first search strategies [ZAZW12] as discussed below.

Gollapudi and Sharma [GS09] show that many diversification objectives can be expressed as obnoxious facility dispersion optimization problems, i.e., the task to place facilities as far away from each other under given constraints and other optimization criteria. Building on this approach, Zuccon et al. show in [ZAZW12] that MMR, the Modern Portfolio Theory (MPT, [WZ09a]), and the Quantum Probability Ranking Principle [ZA10] can be expressed as facility dispersion problem, as well.

Additionally, Zuccon et al. [ZAZW12] introduce desirable facility placement (DES) to express a diversity objective. Desirable facility location is the task to locate a given amount of facilities such that the average distance of a customer to the nearest facility is minimized. This problem can be easily mapped to the information retrieval task by viewing the result documents as facility locations, and the user's information needs as customer locations. The reported gain in retrieval effectiveness for DES is the highest of all diversification approaches proposed so far, motivating us to use DES as base for our work. We give a more in-depth description of DES in Section 3.3.

While the proposed diversity objectives improve the quality of the returned results, diversification comes with a considerable computational penalty. Carterette [Car09] has shown that optimal diversification is NP-Hard. Consequently, for practical use approximations and heuristics need to be used for computing a diverse result set. But even in this case, most proposed algorithms have a quadratic complexity, rendering them infeasible for the demanding efficiency constraints of online query processing. The only exceptions we are aware of are [MSN11] and [DP09] that compute diverse set approximations over continuous data. Our aim is to overcome this limitation and propose a highly efficient diversification approach.

3.2.2 Explicit Result Diversification Techniques

In the explicit diversification methods, query aspects are modeled explicitly by exploiting some sort of external knowledge. IA-Select method proposed by Agrawal et al. [AGHI09] assumes that both queries and documents are associated with some categories from a taxonomy, and then achieves diversification by favoring documents from different categories, and penalizing the documents that fall into already covered categories.

Radlinski and Dumais [RD06] use reformulations of the given query to determine the candidate result set; i.e., the candidate set is formed as the union of results for the original query and its reformulations, and then re-ranked based on the interests of the current user. The xQuAD framework by Santos et al. [SMO10a] also exploits query reformulations obtained from TREC subtopics and search engines, and achieves diversification by identifying the relevance of candidate documents to these sub-queries and favoring documents that cover those aspects not yet covered in the current result set. This approach is found to be top performer in earlier TREC campaigns (e.g., [CCS09, CCSC10]). In [SMO11], both xQuAD and IA-Select are employed in an intent-aware diversification framework, which specifically takes into account the user query intent (such as navigational, informational, etc.). Chapelle et al. [CJL$^+$11] also takes into account query intents that are defined within a shopping scenario, and develops a specific ranking function per intent. Capannini et al. utilize query logs to decide when and how query results should be diversified, and present the new algorithm OptSelect that takes into account the popularity of query reformulations in the log [CNPS11]. Vargas et al. [VCV12] propose to reformulate xQuAD and IA-Select by incorporating a formal relevance model. Vallet and Castells [VC12] further extend the latter two approaches to obtain personalized diversification of search results.

Being inspired from the electoral process used in some countries, Dang and Croft [DC12] introduce a novel explicit strategy that takes into account proportionality of the votes given to the query aspects. In a follow-up work, Dang and Croft [DC13] further argue that an explicit aspect need not to be represented in the form of a set of terms, but considering each such term as a separate aspect is equally useful, or even better. Wu and Huang [WH14] propose to combine multiple retrieval results (from different systems) for a given a query with the expectation of having a more diversified list at the end. Liang et al. [LRdR14] leverage a similar idea, but in their work, after the data fusion stage, the latent aspects are discovered by the LDA algorithm that also take into account the retrieval scores of the fused list. Finally, the combined list and discovered aspects are all fed to the explicit diversification algorithm of Dang and Croft [2012]. Note that, theirs is not considered as an implicit approach as the initial data fusion stage operates over the result lists that have been already diversified using some explicit strategy. In contrast to the latter two approaches that fuse different retrieval results for the *same query*, Ozdemiray and Altingovde [OA15] adopt score-based (such as CombSUM and CombMNZ) and rank-based (such as Borda voting and fusion approaches with Markov chains) aggregation methods to merge the multiple re-rankings of candidate documents for *different query aspects*. Ozdemiray and Altingovde [OA14] also propose strategies to determine the aspect weights during explicit diversification, and present gains in diversification effectiveness of almost all of the state-of-the-art explicit methods.

Note that while explicit diversification approaches are more effective than implicit methods (e.g., see [CNPS11]), the best performing methods (e.g., [SMO10a, CNPS11]) essentially exploit external sources of information such as query logs, and can identify a query aspect only after the reformulations appear in the logs. This works well for popular queries, but queries in the long tail of the popularity distribution may not benefit from such diversification approaches [CJL$^+$11]. In such cases, implicit diversification methods are still applicable, and this underlies our motivation in this work to improve the efficiency of one of the most effective implicit strategies based on DES and GLS, as we describe in Section 3.

3.2.3 Distributed Search on Computing Clusters

Nowadays, the usage of computing clusters with thousands of nodes for processing of search requests on large document collections is firmly established. In such an infrastructure, the index needs to be partitioned. In document-based partitioning, each node in a cluster of servers is responsible for an index that is created for a particular subset of documents. In term-based partitioning, each node stores the fragment of an index corresponding to a subset of terms in the collection. The former approach is usually preferred by large-scale systems [Dea09], due to its lower usage of resources (like network bandwidth) and better load balancing.

In a document-based partitioning architecture, the query processing is said to be embarrassingly parallel. In the basic query processing workflow (e.g., [OAC$^+$12]), the processing task is split up between broker nodes which coordinate query processing, and search nodes which hold index partitions. First, the search engine front-end forwards the query to one of the broker nodes in the search cluster. The broker node in turn distributes the query to the search nodes that access their index files, compute partial top-k results and send them back to the broker. Then the broker determines the global top-k results and contacts document servers to obtain snippets for result presentation. Note that documents can reside at the search nodes or in a separate set of servers [Dea09]. Similarly, it is also possible to let each physical node function as both a search node and broker [OAC$^+$12].

While several aspects of query processing are investigated over the distributed search architecture (such as query forwarding [CVK$^+$10], caching [OAC$^+$12], etc.), to the best of our knowledge, no previous study addresses how online result diversification algorithms can operate on such an architecture and how their results might be affected according to the layer where the diversification takes place.

ALGORITHM 4: GLS: Diversification as DES using Greedy Local Search [ZAZW12]

```
Input : $D, k, f$
Output: $S$
$S \Leftarrow \{d_1, \cdots, d_k\}$
repeat
    for $d \in S$ do
        for $d' \in D \backslash S$ do
            $S' \Leftarrow (S \backslash \{d\}) \cup \{d'\}$
            if $f(S') < f(S)$ then
                $S \Leftarrow S'$
            end
        end
    end
until $S$ does not change
```

3.3 Efficient Greedy Local Search for DES

As we also outlined in the previous section, casting the problem of search result diversification as desirable facility placement (DES) problem from Operations Research yields high-quality retrieval results [ZAZW12]. In this case, the driving motivation is that each document in the top-k search results represents a facility at a different region and every customer (i.e., each possible query intent) should be sufficiently close to one of these facilities. This goal is achieved by choosing the facility locations (i.e., the top-k documents) such that the total distance to all other documents is minimized. As usual for diversification, this coverage criterion is balanced with document relevance to determine the final result set. Formally, the optimal set of top-k results for a query q is given by:

$$S^* = \underset{\substack{S \subset D \\ |S|=k}}{\arg\min} f(S) \tag{3.1}$$

$$f(S) = -\lambda \sum_{d \in S} r(d) + (1-\lambda) \sum_{d' \in D \backslash S} \left(\min_{d \in S} w(d, d') \right) \tag{3.2}$$

where D is the set of candidate documents (i.e., an initial retrieval result for q), $f(S)$ the diversification objective function, $r(d)$ the relevance score of document d for query q, and $w(d, d')$ the distance between two documents d and d'.

In [ZAZW12], Zuccon et al. proposed an approximate solution for computing DES using Greedy Local Search (GLS, see Algorithm 4). GLS starts with placing the most relevant k documents of N candidate documents (denoted with set D) into the set S, i.e., top-k search results. At each round of the algorithm, the documents in S are replaced with documents that are not in S, to optimize the objective function

ALGORITHM 5: Objective Function for C-GLS

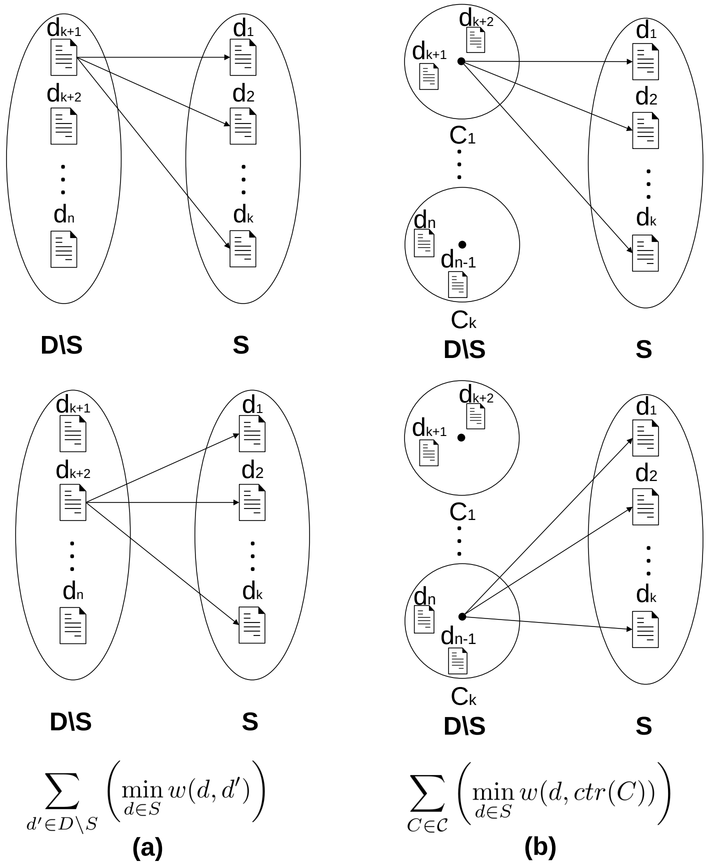
```
Input   : $S, \lambda$
Output: score
relscore ⇐ divscore ⇐ 0
for $d \in S$ do
    relscore ⇐ relscore $- \lambda \mathtt{r}(d)$
end
for $C \in \mathcal{C}$ do
    minDist ⇐ 1
    for $d \in S$ do
        minDist ⇐ $\mathtt{min}$(minDist, $\mathtt{w}(ctr(C), d)$)
    end
    divscore ⇐ divscore + $(1 - \lambda)$minDist
end
score ⇐ relscore + divscore
```

score. If the latter score does not change anymore, the algorithm terminates, as it has reached a global or local minimum. The authors have also shown that framing result diversification approaches such as MMR, MPT and QPRP into DES and using the GLS approximation is quite effective, and indeed yields considerably better diversification performance than applying the traditional Best First Search heuristic.

While exhibiting an impressive diversification performance, GLS is computationally expensive. Each round of Algorithm 4 requires swapping each document in S with each document in $D \backslash S$, which means $k \times N$ calls for computing the objective function, $f(S)$. As shown in Equation 3.2, the computation of $f(S)$ also requires finding the closest document in S to each document in $D \backslash S$, and this costs another $N \times k$ operations to compute pairwise distances. Thus, in total, the computational complexity of a single round of the algorithm is $O(N^2k^2)$. Assuming that the algorithm converges in I_{GLS} rounds, the overall complexity is $O(N^2k^2I_{GLS})$.

Note that if the distance $w(d_i, d_j)$ is computed on the fly, this cost would further increase by the complexity of the document similarity measure (e.g., for the cosine-based similarity $O(\min(|d_i|, |d_j|))$. Therefore, we assume that all pairwise distances among the candidate documents (typically top-100 or top-1000) are computed once (in a preprocessing stage) and then cached till the end of processing (as in [MSN11]). In Table 3.1, we provide the break-up of CPU and memory consumption for this latter case (i.e., with cached distances). Given that there is no bound on the convergence of the algorithm and it might take many rounds until it converges (as we illustrate in our experimental evaluations), the algorithm is expensive for practical scenarios.

Clustering-GLS (C-GLS) algorithm. In this work, we propose to improve the efficiency of the GLS solution by employing clustering to approximate the distance computation stage for the objective function computation (see Algorithm 5).



$$\sum_{d' \in D \setminus S} \left(\min_{d \in S} w(d, d') \right)$$

$$\sum_{C \in \mathcal{C}} \left(\min_{d \in S} w(d, ctr(C)) \right)$$

Figure 3.1. Objective function computation for a) GLS (i.e., for each document d' in $D\setminus S$, we compute the distance $w(\bullet)$ to each document d in S), and b) Clustering-GLS (i.e., for each cluster centroid C in $D\setminus S$, we compute the distance to each document d in S).

In particular, we form a clustering of the documents in D such that each document falls into a single cluster C with centroid $ctr(C)$. Instead of storing pairwise distances, $w(d_i, d_j)$, we store for each document the distance to all cluster centroids, $w(d_i, ctr(C_j))$. Then, while computing the distance of documents that are in $D\setminus S$ to those in S, instead of considering every single document in $D\setminus S$, we only consider cluster centroids, i.e., as an approximation of the document space $D\setminus S$ (compare Figure 3.1(a) and (b)). As our result set can represent at most k different query intents, it is sufficient to split D into k different clusters C_i. With $\mathcal{C} = \{C_1, \ldots, C_k\}$ the cluster-based objective function becomes

$$f(S) = -\lambda \sum_{d \in S} r(d) + (1-\lambda) \sum_{C \in \mathcal{C}} \left(\min_{d \in S} w(d, ctr(C)) \right) \tag{3.3}$$

As Algorithm 5 shows, computing this objective function has a complexity of $O(k^2)$ instead of $O(k \cdot N)$. Since the objective function is called $k \times N$ times, the overall complexity of the algorithm becomes $O(Nk^3)$ (per round).

In the clustering stage, we essentially employ k-means clustering algorithm. This is indeed is a natural choice for GLS problem; as already pointed out in [ZAZW12], for $\lambda = 1$ the objective function (Eq. 3.2) leads to k-medoids clustering of D as a result. In our evaluations, we also experiment with a single-pass approach based on the *list of clusters* idea introduced by [CN05] and adopted for diversification by Gil-Costa et al. [GCSMO13].

Clustering^2-GLS(C^2-GLS) algorithm. When choosing reasonable λ values, i.e., values which lead to a balance between relevance and diversity, it can be observed that documents with low relevance score don't end up in the final top-k result set, regardless of the chosen DES variant. In a sufficiently large document collection, it is safe to assume that for several documents matching a specific query intent, the more relevant ones will be preferred (although such a selection might not lead to the perfect result set composition). It is therefore very ineffective to try replacement candidates d' at random (Alg. 1, Line 6); instead, we should identify the most promising candidates for each query intent, and limit the greedy search procedure to this selected set.

We exploit clustering *again* to identify these candidates as well as using in the objective function shown in Algorithm 5 (and hence we call this strategy C^2-GLS). We sort the documents of each cluster by relevance and define our candidate set $TopC$ as the union of the top-r documents from each cluster C_j. By taking only these most relevant documents into account and with $R = |topC|$, we achieve a complexity of $O(Rk^3)$ (per round), which is a further improvement over $O(Nk^3)$ given that $R << N$.

Table 3.1 shows the cost break-up of so called Clustering-GLS (C-GLS) and Clustering^2-GLS (C^2-GLS) strategies with k-means clustering algorithm. The pairwise distance computation cost now reduces to $O(NkI_C)$ where I_C is the number of iterations for k-means clustering. Note that, the clustering algorithm is expected to converge in a few iterations as N is at most a few thousands in practical settings. More crucially, since the computation of $f(S)$ now involves comparison of cluster centroids to top-k documents, it has $O(kN)$ complexity, reducing the overall complexity of Algorithm 4 from $O(N^2k^2)$ to $O(Nk^3)$ and $O(Rk^3)$ (per round), for C-GLS and C^2-GLS strategies, respectively. Given that k is usually one or two order of magnitudes smaller than N (i.e., $k = 10$ while N is either 100 or 1000), this is a crucial improvement in terms of efficiency. Furthermore, our C-GLS and C^2-GLS strategies have a memory foot print linear in the document collection size, i.e., $O(Nk)$ instead of $O(N^2)$, as we only need to store for each document the distance to the cluster centroids. Last but not the least; the use of cluster centroids also induces a smoothening effect on the distance values, and the proposed algorithms converge in a smaller number of rounds (as shown in the experiments).

Table 3.1. Complexity of diversification algorithms (I_C denotes the number of rounds for clustering; CPU complexity costs for the actual diversification stage are in terms of the number of distance computations *per round*)

Algorithm	CPU (Preprocessing)	CPU (Diversification)	Memory
GLS	$O(N^2)$	$O(N^2k^2)$	$O(N^2)$
C-GLS	$O(NkI_C)$	$O(Nk^3)$	$O(Nk)$
C^2-GLS	$O(NkI_C)$	$O(Rk^3)$	$O(Nk)$

3.4 Distributed Diversification

While the effectiveness and efficiency of the diversification algorithms have received serious attention in the literature, to the best of our knowledge, the architecture over which these algorithms would be employed has not been considered. In this section, we introduce two distributed diversification approaches and discuss their pros and cons in a realistic setup.

In practice, all large-scale search engines operate on a number of geographically distributed computing clusters, each with tens of thousands of servers (nodes)[1]. Each node in a cluster stores an inverted index that corresponds to a randomly partitioned subset of entire collection, as well as other data statistics (such as document lengths, etc.) required for query processing. A search cluster has one or more broker nodes that are responsible for forwarding the query to all search nodes, gathering the top-k partial results from these nodes and merging the partial results to obtain global top-k results.

In this section, we investigate diversification of search results in a typical search cluster as described above. We envision that result diversification for a given query can take place either in the broker node or in the indexing nodes and define corresponding strategies as follows:

Broker-based diversification (BB-Div). This is the straightforward case where the broker runs a diversification algorithm once it collects and merges all the partial results from the search nodes (see Algorithm 6). Assuming that each of the P nodes returns top-k (partial) results computed on its local collection D_P, the broker will have a set of $P * k$ documents. Then, it can apply the diversification algorithm on these $P * k$ documents, or further restrict this initial set to top-N results with the highest relevance scores (for the sake of efficiency). We opt for the latter option as it is more likely in practice (otherwise, for a typical cluster of fifty thousand servers, the candidate set would be huge). Besides, as we discuss in the next section, fixing the

[1] In this work, we focus on distributed query processing within a single search cluster that can usually capture a replica of the entire Web index.

ALGORITHM 6: BB-Div

```
Node    $D_{P,k} \Leftarrow Top(D_P, k)$              // Compute local top-$k$ result
Broker  for $P \in \mathcal{P}$ do
            $D \Leftarrow D \cup D_{P,k}$                // Merge top-$k$ from nodes
        end
        $D \Leftarrow \mathtt{Top}(D, N)$                   // Keep top-$N$ results
        $S \Leftarrow \mathtt{Diversify}(D, k)$
```

candidate set size allows a fair comparison of the distributed diversification approaches in our simulations. Thus, in Algorithm 6, the broker first calls the function $Top(.)$ to construct the candidate set D of size N, and then invokes the diversification algorithm to obtain the final result set S (of size k). Note that, if the diversification algorithm imposes an order on the documents selected into S (e.g., by generating a score), the results are presented in this order. Otherwise, for the algorithms (like DES) where the output S is a set but not a ranked list, the final top-k results are still presented in the order of their initial relevance scores. The BB-Div strategy is illustrated in Figure 3.2 (a) for a toy scenario with the latter assumption. In this scenario, d_1 and d_4 are sent to the broker as they achieved the highest relevance scores at their nodes; and at the end of the diversification applied at the broker, we assumed both appeared in the final result, because they are both relevant and also different from each other (as denoted by different color codes and textures in the figure).

The advantage of the broker-based diversification strategy lies in its simplicity and practicality; it can be directly coupled with an existing search system. On the other hand, there are a couple of drawbacks that should be taken into account in terms of the efficiency and effectiveness of the diversification: First, as each server returns its top-k results (where k is usually 10), the candidate list to be diversified can miss results that are related to different interpretations of a query, especially if a particular interpretation dominates the result set. For instance, for the query *Java*, partial top-10 results from each node can rank only those documents related to Java as a programming language, so that diversification in the broker by setting N to 100 or 1000 (or, even using all $P * k$ results) might be sub-optimal, simply because the candidate set is not large enough to encounter the documents that cover different query intents. We illustrate this case also in Figure 3.2 (a): the third node returns its top-2 results (d_7 and d_8), both of which have the same pattern/color, and hence, misses the third result, d_9, which is different from the first two documents. This might be remedied by increasing the partial result set size, if the query can be identified as ambiguous beforehand, which is a non-trivial issue on its own (e.g., see [CNPS11]).

As a further yet related efficiency problem, the broker-based diversification requires the document vectors for top-N documents to be transferred to the broker node, as most of the implicit and explicit diversification approaches need the document vectors. For instance, all implicit diversification approaches we focus on in

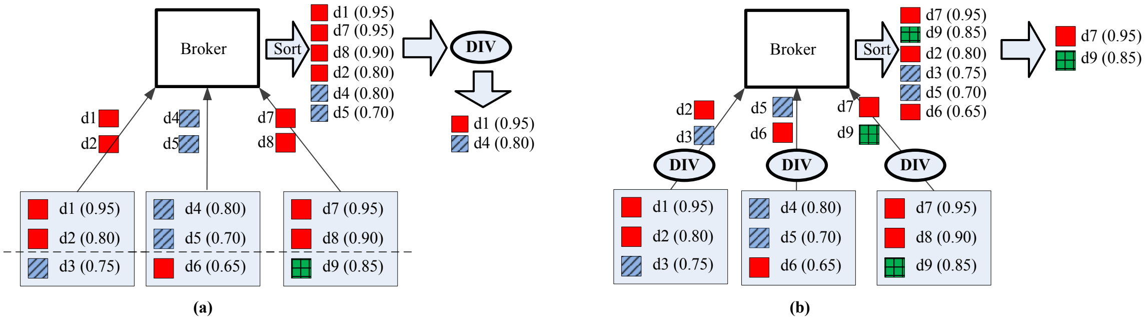


Figure 3.2. A toy example for (a) Broker-based, (b) Node-based diversification strategies (All nodes and the broker return top-2 results). Note that the final diversified results differ.

this work would need vectors for pairwise distance computations. Indeed, even for an explicit diversification technique like xQuAD [SMO10a], the document vectors may still be required to compute the similarity of explicit query intents to each candidate document in the broker[2]. Moving document vectors among the nodes incurs some network cost even if each document resides in a different node and transfers can be processed in parallel. This cost would further increase for larger values of N and/or partial result set size. In the next section, we provide a detailed analysis of the network cost for this scenario. Note that, once document vectors are transmitted to the broker, the pairwise distances (or, coverage of explicit query aspects) have to be computed on the fly, since the top-N candidate documents will be almost always compiled from different nodes (due to partitioning of the collection uniformly at random to the nodes) and hence, these distances cannot be computed a-priori.

Node-based diversification (NB-Div). An alternative strategy is applying the result diversification in each search node and combining the partial top-k results that are already diversified (see Algorithm 7). In this case, each of the P nodes first obtains the candidate set $D_{P,N}$ (of size N) on its local collection D_P, and then calls the diversification algorithm to select the diversified top-k set into $S_{P,k}$. The generated partial results are simply merged at the broker based on their relevance scores (or, diversification scores, if available), to create the global top-k answer. This case is illustrated in Figure 3.2 (b). In this case, the top-ranked document d_1 is eliminated at the node its hosted, as its successors in the list, d_2, which has a lower relevance score, is assumed to have a much higher dissimilarity to the third document in the list (d_3) (to illustrate how NB-Div differs from BB-Div), and hence, these two documents, d_2 and d_3 are sent to the broker.

As the indexed documents can be stored along with the index at each node [OAC+12], node-based diversification strategy has no transfer costs to access the documents.

[2] Alternatively, each query aspect has to be sent back to the nodes and the relevance scores for each such aspect and every document in the candidate set need to be computed using the inverted index. We further investigate the efficiency of this approach in the next section.

ALGORITHM 7: NB-Div

```
Node    $D_{P,N} \Leftarrow$ Top($D_P, N$)
        $S_{P,k} \Leftarrow$ Diversify($D_{P,N}, k$)
Broker  for $P \in \mathcal{P}$ do
            $S \Leftarrow S \cup S_{P,k}$              // Merge Div-$k$ from nodes
        end
        $S \Leftarrow$ Top($S, k$)                    // Keep top-$k$ results
```

Furthermore, similarities between a certain subset of documents (such as those with the highest PageRank scores or highest frequency in the query results) can be precomputed and cached, to be used for different queries. This strategy also allows considering a deeper pool of documents while still executing the diversification algorithm for top-N candidates. That is, for a fixed value of N, the cost of running a diversification algorithm at the broker (excluding the document transfer times as discussed above) is the same as running the same algorithm at the nodes, but since each node has its own top-N set, it is more likely to encounter documents with diverse interpretations (e.g., in the *Java* example above). However, this might also work in the reverse direction: when each partial result set is locally diversified, some of the results in partial top-k can indeed have a very low global rank in terms of relevance, i.e., some potentially more relevant results might be sacrificed at each node for diverse yet very low ranked results. This is an interesting trade-off that might require adaptive solutions based on the query properties, such as the degree of ambiguity [SMO10b].

A particular disadvantage of this algorithm is the lack of global knowledge during the diversification stage. That is, since each list is diversified locally, the relative order of diverse intents might be similar (e.g., all top-1 documents from each node might be relevant for the programming language intent for the *Java* query) and the simple relevance-based merging at the broker can easily end up with less diversification than desired.

In the following section, we evaluate both the GLS based implicit diversification algorithms and xQuAD, a well-known explicit diversification algorithm, in a distributed environment, observe how the aforementioned trade-offs affect their effectiveness and efficiency, and draw conclusions for designing and employing diversification algorithms in large-scale search systems.

3.5 Experiments and Results

3.5.1 Experimental Setup

Document collection. In this work, following the practice in the diversification field, we use one of the largest available Web datasets with expert assessments, namely, the Clueweb09 Part-B collection[3] that includes around 50 million Web pages. We index the collection using the Zettair IR system[4], with the "no stemming" option. All stop-words and numbers are included in the index, yielding a vocabulary of around 160 million terms.

Queries and initial retrieval. We use the query topics and relevance judgments released for TREC 2009 and 2010 Diversity Task[5] that include 50 and 48 topics, respectively. For each topic, there are a number of pre-defined subtopics (between 1 and 8) that are used during the judgment process; so that each document annotated as relevant is also associated with one or more subtopics from this list. The TREC queries are generated automatically using the title fields of the topics.

Additionally, we employ a third and larger set (denoted as *Q1000*) that includes 1,000 queries that are sampled from AOL 2006 query log [PCT06] and can retrieve non-empty results over our collection. This latter set is only used for evaluating efficiency but not the effectiveness, as there is no available relevance judgments for the queries in this set. A similar approach is also followed in [GCSMO13] for evaluating diversification efficiency.

We used our own IR system (also employed in [OA15]) to process the queries over the index. As the relevance model, we use a variant of the well-known Okapi-BM25. We set the free parameters k_1 and b to 1.2 and 0.75, respectively.

Document similarity computation. We employ the usual Cosine similarity of document vectors with tf-idf weights for computing the similarity $s(d, d')$ between two documents d and d'.

Diversification algorithms. In addition to GLS and its proposed variants C-GLS and C^2-GLS methods, we involve List of Clusters Diversification (LCD) and xQuAD as further baselines from the implicit and explicit diversification literature, respectively. We summarize their implementation and parameters as follows.

GLS,C-GLS and C^2-GLS. As discussed in the previous sections, we focus on desirable facility placement (DES) approaches and evaluate three approximate solutions: GLS from [ZAZW12], C-GLS and C^2-GLS. In [ZAZW12], it is also shown that well-known diversification techniques like MMR, QPRP and MPT can be all modeled within the DES framework, by choosing corresponding relevance and distance measures in Equation 3.2. Their experiments further reveal that the best performing strategy in the DES+GLS framework is MPT, outperforming MMR and QPRP. Therefore, in this work, we instantiate the objective function in Equations 3.2 and 3.3 for MPT approach as in [ZAZW12] for all three diversification algorithms. So, $r(d)$ is set to document's BM25 score (normalized per query by dividing relevance scores by the maximum BM25 score for a given query) and

[3] http://www.lemurproject.org/clueweb09.php/

[4] www.seg.rmit.edu.au/zettair/

[5] http://trec.nist.gov/data/web09.html

$$w(d, d') = 2 \times b \times \sigma^2 \times w_{d'} \times (1 - s(d, d')) \tag{3.4}$$

where $w_{d'}$ is the importance weight of the rank of d' in S, computed in the same fashion to discounting factors of nDCG metric [JK02b], and b and σ^2 are treated as parameters for MPT following the practice in [ZAZW12]. We experiment with values of b in the range $[1, 10]$ with increments by 1, and σ^2 in the range $[10^{-6}, 10]$ incremented by orders of 10. Finally, for our C^2-GLS algorithm, we set r to 5; i.e., we consider top-5 most relevant documents of each cluster in the algorithm.

LCD. As a further baseline, we involve another implicit approach, namely, the List of Clusters Diversification (LCD) algorithm introduced by Gil-Costa et al. [GCSMO13]. As discussed in the related work section, this latter work also aims to improve the efficiency of implicit diversification and proposes three different methods that utilize metric spaces for efficient computation of the pairwise distances. Among these methods, we choose LCD as baseline due to two reasons. First, the LCD method applies clustering for diversification, so it is methodologically close to our C-GLS and C^2-GLS methods that also employ clustering, albeit as a pre-processing stage. Second, according to their experimental results, when the underlying retrieval model is BM25 (which is also the case in this work), the best-performing method is LCD (please see Table 1 in [GCSMO13]).

In a nutshell, the LCD algorithm works as follows. The document with the highest retrieval score is selected as the center of the first cluster; and the distance of all the other documents to the center is computed. Then, a fixed number, say r, of the documents that are nearest to this first center are assigned to its cluster, and removed from the candidate set. The next cluster center is chosen as the one that maximizes the sum of distances to the previous center(s), and again, the r-nearest documents are assigned to this cluster. The process continues until all the candidate documents are clustered and results in a List of Clusters (LC) [CN05]. Finally, LCD algorithm ranks the cluster centers at the top of the final result list (in the order of the discovery), and then lists their members as blocks (in the order of relevance) in the corresponding order of their centers. In our implementation of LCD, we set the cluster size, r, as the value that yields the highest diversification performance for each query topic set.

xQuAD. While our cluster-based solutions are intended to improve the efficiency of GLS as an effective implicit diversification algorithm; the strategies proposed for a scalable distributed architecture are independent of the diversification algorithm; i.e., any kind of diversification approach can be incorporated into BB-Div and NB-Div strategies. Therefore, we do not restrict our evaluations over the distributed architecture to only implicit diversification algorithms, but also employ xQuAD as a further baseline. We believe that xQuAD is a good representative of the class of explicit diversification algorithms, as it is placed among the top performers in the diversity tasks of TREC from 2009 to 2012. Furthermore, most recent explicit diversification algorithms, namely, IA-Select [AGHI09], xQuAD [SMO10a], PM2 [DC12], and ranking aggregation based methods proposed in [OA15], all need to compute the relevance scores of the candidate documents for the query aspects, which means that they would incur the same cost in terms of the network communication in a distributed setup. Hence, xQuAD serves as an adequate representative also from the latter perspective.

We implemented xQuAD following the common practice in the earlier works [SMO10a, DC12, OA15] to simulate an ideal setup, i.e., with the perfect knowledge of the query aspects. To this end, explicit query aspects are generated using the official TREC sub-topic descriptions for each topic. In our experiments, we test all values of the trade-off parameter λ in [0,1] range with a step size of 0.01, and report the results for the λ that optimizes α-NDCG@20. Note that, in a more realistic scenario, the query aspects could be obtained from the query suggestions of a search engine [SMO10a] and λ values could be learned within a machine learning setup as in [SMO10b]. However, we prefer to use the best-performing setup for xQuAD as our goal here is not comparing the effectiveness of implicit and explicit diversification algorithms; but providing an in-depth investigation of how these algorithms perform when they are incorporated into our distributed diversification strategies.

Clustering algorithms. As discussed in Section 3, we essentially employ the standard k-means clustering algorithm for the preprocessing in C-GLS and C^2-GLS methods. As a further baseline, we employ the LC approach [CN05] described before in the context of LCD, as it is a single-pass algorithm and found to be efficient when the target number of clusters is small [GCSMO13].

Evaluation metrics. We use the evaluation software *ndeval* provided as part of the TREC Diversity Task. We report effectiveness results using α-NDCG [CKC$^+$08a], ERR-IA [CMZG09a] and sub-topic recall (S-recall) [ZCL03], which are widely used in the literature. To evaluate efficiency, we report the elapsed time for the preprocessing (i.e., pairwise distance computations for GLS, and the cost of k-means or LC clustering for the proposed methods C-GLS and C^2-GLS) and actual diversification stages, per query. To facilitate the reproducibility of our results by others, we also report machine-independent measures, namely, the average number of rounds till convergence, average number of times for invoking the objective function (in each round), and average number of look-ups for pairwise distances (between two documents or between a document and a cluster centroid) in each call of the objective function. Finally, for the distributed diversification setup, we evaluate the performance in terms of the network communication cost, namely, total volume of the transferred data (in bytes) and transfer time (in milliseconds).

3.5.2 Evaluation of the C-GLS and C^2-GLS

As our first goal is improving the efficiency of GLS solution for DES approach in result diversification, we compare the effectiveness and efficiency of GLS to C-GLS and C^2-GLS strategies that are proposed in this work. In this set of experiments, we assume a single node architecture as typical in the literature; i.e., for a given query, we retrieve top-N documents (D set) from the entire collection and then re-rank these candidate documents to obtain the final top-k results (S set) using GLS, C-GLS and C^2-GLS with MPT instantiation. We set $k = 20$ and $N = 100$.

Effectiveness Evaluation. In Table 3.2, we compare the effectiveness of the baseline and proposed diversification algorithms and the standard BM25 baseline, i.e., retrieving top-20 most relevant documents without any diversification, for TREC 2009 and 2010 topic sets. First of all, we observe that the non-diversifed baseline using BM25 is much better than the language model baseline reported in [ZAZW12]. For instance, while NDCG@5, @10 and @20 are found to be 0.105, 0.150 and 0.207 in [ZAZW12], our baseline yields 0.183, 0.217 and 0.250, respectively. Capannini et al. also report values closer to ours, namely, 0.190, 0.212 and 0.240 respectively for NDCG@5, @10 and @20 while they have employed Divergence From Randomness (DFR) model for the retrieval [CNPS11]. We believe that these differences can be caused by different choices that might be made during the document processing, indexing and/or query processing, as each of these stages involve several parameters that can affect the final result.

Table 3.2 shows that xQuAD, in its best-performing setup, can beat both the standard baseline and implicit diversification approaches, a finding that confirms the previous results in the literature. As we have pointed out, our goal in this work is improving the efficiency of GLS as an implicit diversification solution, as such approaches are viable/helpful for a wide range of practical cases where the query aspects cannot be known in advance. Therefore, here we provide the effectiveness values for non-distributed xQuAD only for reference, to enable the analysis of its performance within the distributed framework in the following section.

For the implicit strategies, our experiments reveal that GLS can outperform the standard baseline, although the gains are not as pronounced as those reported in [ZAZW12]. In contrast, the alternative implicit diversification baseline, LCD, can improve the non-diversified BM25 ranking only for a couple of cases for TREC 2010 topic set. Note that, the latter finding is not far from the earlier results reported in Table 1 of [GCSMO13], where an improvement of at most 0.0067 is observed for LCD using similar metrics at the cutoff value 20. These results also justify our goal of improving the efficiency of GLS in this work, due to its high effectiveness as an implicit diversification algorithm.

Our proposed methods, C-GLS and C^2-GLS, are evaluated using two different clustering algorithms, namely, k-means and LC. Table 3.2 shows that, especially for the cases with the k-means algorithm, our methods have no adverse impact on the diversification effectiveness and indeed, at almost all rank cutoffs, both of the proposed strategies are better than the baseline and perform comparable to (or, especially for the TREC 2010 set, even better than) the GLS algorithm. The latter finding on the TREC 2010 dataset can be explained by the smoothing effect of the clustering strategies, i.e., our algorithm avoids considering very diverse candidates with very low relevance scores. We also observe that using k-means for the preprocessing yields

Table 3.2. Retrieval effectiveness of the diversification algorithms. Type field denotes implicit or explicit diversification. The superscripts ($*$) and ($\dagger$) denote a statistically significant difference at 0.05 level from the baseline and GLS algorithms, respectively. The xQuAD algorithm that utilizes explicit knowledge of aspects is included only for reference, to be considered in the evaluation of the distributed framework.

			ERR-IA			α-NDCG			S-recall		
TREC 2009											
Algorithm	Preproc.	Type	@5	@10	@20	@5	@10	@20	@5	@10	@20
Baseline		None	0.120	0.134	0.142	0.183	0.217	0.250	0.256	0.346	0.414
LCD		Imp.	0.120	0.134	0.142	0.183	0.217	0.250	0.256	0.346	0.414
GLS		Imp.	0.150*	0.162*	0.168*	0.228*	0.248	0.269*	0.286*	0.347	0.391
C-GLS	k-Means	Imp.	0.155*	0.164*	0.171*	0.233*	0.248*	0.272*	0.316*	0.353	0.420
C^2-GLS	k-Means	Imp.	0.153*	0.163*	0.169*	0.230*	0.247*	0.270	0.313*	0.381	0.435
C-GLS	LC	Imp.	0.156*	0.169*	0.175*	0.227*	0.250*	0.276*	0.299*	0.351	0.424
C^2-GLS	LC	Imp.	0.146*	0.158*	0.165*	0.220*	0.244*	0.270	0.291	0.368	0.426
xQuAD		Exp.	0.158	0.175	0.183	0.226	0.264	0.298*	0.288	0.390†	0.460*†

			ERR-IA			α-NDCG			S-recall		
TREC 2010											
Algorithm	Preproc.	Type	@5	@10	@20	@5	@10	@20	@5	@10	@20
Baseline		None	0.141	0.156	0.165	0.182	0.214	0.245	0.276	0.377	0.452
LCD		Imp.	0.141	0.156	0.165	0.182	0.214	0.249	0.276	0.377	0.478
GLS		Imp.	0.138	0.156	0.166	0.184	0.221	0.255	0.276	0.386	0.483
C-GLS	k-Means	Imp.	0.168*†	0.187*†	0.195*†	0.209*†	0.249*†	0.278*†	0.287	0.408	0.507*
C^2-GLS	k-Means	Imp.	0.164*†	0.175	0.184*	0.210†	0.231*†	0.265*†	0.310	0.385	0.506
C-GLS	LC	Imp.	0.151*	0.162*	0.169*	0.208*	0.233*†	0.266*†	0.278	0.356	0.432
C^2-GLS	LC	Imp.	0.159*†	0.176*†	0.185*†	0.202*†	0.238*†	0.270*†	0.298	0.395	0.502
xQuAD		Exp.	0.177	0.194	0.200	0.234	0.272*†	0.293*†	0.390*†	0.488*†	0.526*†

higher effectiveness than using LC, for the majority of the cases (e.g., C^2-GLS with LC is inferior to the version with k-means for the TREC 2009 set for almost all metrics). This is also expected, as the multi-pass nature of the k-means algorithm may yield a better clustering of the candidate documents.

Impact of the parameters N and k. We also analyze the sensitivity of our methods for the candidate result set size (N) and the number of clusters (k). For the former parameter, earlier studies using a similar TREC setup report that smaller values (such as 50 or 100) are better [DC12, DC13, OA15]. A possible explanation for this observation is that the documents that are ranked too low are more likely to be irrelevant yet diverse, and hence their inclusion in the final result reduces the effectiveness. In our case, we also experimented for $N \in \{50, 100, 200, 500, 1000\}$ and found out that a candidate set of 100 consistently yields the best scores for more than half of the cases on TREC 2009 and 2010 sets for GLS, as well as the C-GLS and C^2-GLS methods (with the k-means clustering).

Table 3.3. Diversification performance (α-NDCG@20) vs. the number of clusters (k)

Topic set	Algorithm	Preprocessing algorithm	No. of clusters (k) 5	10	15	20	25
TREC'09	C-GLS	k-means	0.266	0.266	0.270	**0.272**	0.267
	C^2-GLS	k-means	0.263	0.270	0.258	**0.270**	0.267
	C-GLS	LC	0.274	**0.276**	0.270	**0.276**	0.272
	C^2-GLS	LC	0.259	0.265	0.268	**0.270**	0.258
TREC'10	C-GLS	k-means	0.264	0.247	0.261	**0.278**	0.259
	C^2-GLS	k-means	0.258	0.255	0.256	**0.265**	0.256
	C-GLS	LC	0.265	**0.266**	0.262	**0.266**	0.263
	C^2-GLS	LC	0.258	0.266	0.266	**0.270**	0.260

Another important parameter is the number of clusters, k, that is intuitively set to the final result set size (i.e., 20), as it is possible to represent at most 20 different clusters (and equivalently, intents) in the final query result. We also experimented for $k \in \{5, 10, 15, 20, 25\}$. In Table 3.3, we only report the results in terms of the α-NDCG metric at the cutoff value of 20, as the results for the other metrics exhibit completely similar trends and are discarded for the sake of brevity. These experiments reveal that for the majority of the cases, setting k as 20 yields the best effectiveness score in this setup, a finding that justifies our intuitive choice.

Finally, we investigate the stability of the performance of our methods when the k-means algorithm is initialized differently, due to random selection of the seeds (Note that, this is not an issue for LC as it always chooses the same seeds in the same order in a deterministic manner). To this end, for each algorithm and topic set, we applied the k-means algorithm for ten times, yielding different clustering structures, which are then used in the diversification stage. In Table 3.4, we present the statistics for the diversification performance of each algorithm with these ten clustering structures in terms of the minimum, maximum and average scores for each metric. We also report the effectiveness of the original GLS for each case, which is repeated from Table 3.2 to facilitate the comparison. Our findings show that, the effectiveness scores of the algorithms are stable, as the standard deviation for each metric is very low. Furthermore, a comparison of the average scores to the corresponding GLS row shows that our methods consistently perform as good as the original GLS on the average; even with different clustering structures.

Efficiency Evaluation. Being convinced with the effectiveness of our methods C-GLS and C^2-GLS, we turn our attention to their efficiency, which is our main focus here. We report the preprocessing and diversification costs in terms of the CPU processing time. While doing so, we discard the time for generating the candidate result set (D) and retrieving the document vectors, as these stages have the same cost for all compared approaches. Our implementations are single-threaded and hence

Table 3.4. Statistics of the diversification performance for 10 different clustering structures produced by the k-means. GLS scores are provided for easy comparison.

		ERR-IA			α-NDCG			S-recall		
TREC 2009										
Algorithm		@5	@10	@20	@5	@10	@20	@5	@10	@20
GLS		0.150	0.162	0.168	0.228	0.248	0.269	0.286	0.347	0.391
C-GLS	AVG	0.152	0.165	0.171	0.225	0.248	0.273	0.285	0.350	0.416
	MIN	0.143	0.159	0.166	0.213	0.243	0.269	0.263	0.337	0.380
	MAX	0.161	0.172	0.178	0.239	0.255	0.282	0.316	0.365	0.433
	STDEV	0.006	0.005	0.005	0.009	0.004	0.005	0.011	0.009	0.019
C^2-GLS	AVG	0.154	0.165	0.172	0.225	0.247	0.272	0.291	0.355	0.420
	MIN	0.150	0.161	0.169	0.218	0.239	0.264	0.272	0.335	0.390
	MAX	0.161	0.170	0.178	0.235	0.252	0.283	0.319	0.389	0.446
	STDEV	0.003	0.003	0.003	0.005	0.004	0.005	0.008	0.010	0.017
		ERR-IA			α-NDCG			S-recall		
TREC 2010										
Algorithm		@5	@10	@20	@5	@10	@20	@5	@10	@20
GLS		0.138	0.156	0.166	0.184	0.221	0.255	0.276	0.386	0.483
C-GLS	AVG	0.159	0.174	0.183	0.203	0.234	0.267	0.295	0.385	0.498
	MIN	0.150	0.162	0.169	0.183	0.221	0.254	0.247	0.334	0.432
	MAX	0.171	0.182	0.194	0.219	0.247	0.278	0.342	0.424	0.530
	STDEV	0.005	0.004	0.002	0.009	0.006	0.002	0.026	0.012	0.015
C^2-GLS	AVG	0.166	0.181	0.188	0.211	0.241	0.265	0.312	0.389	0.472
	MIN	0.162	0.175	0.184	0.202	0.231	0.259	0.286	0.373	0.432
	MAX	0.170	0.188	0.194	0.216	0.253	0.271	0.333	0.403	0.506
	STDEV	0.003	0.004	0.003	0.004	0.006	0.003	0.011	0.010	0.006

executed on a single CPU, although we use a server with 31 Intel Xeon processors and a total of 32GB of RAM, and running CentOS Linux 6.6 distribution.

Our results in Table 3.5 reveals that the implicit diversification baseline, LCD, is extremely fast. Indeed, it is even faster than xQuAD, which is reported here only for the sake of completeness, as most explicit approaches have lower computational complexity due to their prior knowledge of the query aspects (e.g., see [OA15]). However, as discussed before, the effectiveness of LCD is only slightly better than the non-diversified BM25 ranking, rendering it rather useless in a realistic setup. For the GLS algorithm with higher effectiveness, the online diversification stage takes more than five hundred milliseconds, which is again impractical.

Fortunately, Table 3.5 demonstrates that our approaches (and especially C^2-GLS) reduce the actual diversification time of GLS by up to three orders of magnitudes (e.g., 747.938 vs. 0.486 ms for GLS and C^2-GLS with k-means on the TREC 2009 topic set, respectively). In terms of the preprocessing time, when we employ the k-means algorithm, the clustering overhead of our approaches seems to be larger than the cost

Table 3.5. Processing time of the diversification algorithms (per query). The last column denotes the improvement over GLS with respect to the total processing time.

Topic set	Algorithm	Preprocessing algorithm	Preprocessing time (ms)	Diversification time (ms)	Total time (ms)	Impr. over GLS
TREC 2009	GLS		72.081	747.938	820.019	-
	LCD	N/A	-	5.604	5.604	99%
	C-GLS	k-means	138.244	38.375	177.149	78%
	C^2-GLS	k-means	138.244	0.486	138.730	83%
	C-GLS	LC	19.104	38.905	58.009	93%
	C^2-GLS	LC	19.104	1.441	20.545	97%
	xQuAD	N/A	-	9.375	9.375	98%
TREC 2010	GLS		92.395	770.687	863.082	-
	LCD	N/A	-	8.175	8.175	99%
	C-GLS	k-means	146.750	37.908	184.658	79%
	C^2-GLS	k-means	146.750	0.481	147.231	83%
	C-GLS	LC	21.314	38.366	59.680	93%
	C^2-GLS	LC	21.314	1.703	23.017	97%
	xQuAD	N/A	-	9.591	9.591	98%
Q1000	GLS		70.100	544.776	614.876	-
	LCD	N/A	-	4.899	4.899	99%
	C-GLS	k-means	132.787	27.123	159.910	74%
	C^2-GLS	k-means	132.787	0.338	133.125	78%
	C-GLS	LC	16.248	27.244	43.492	93%
	C^2-GLS	LC	16.248	1.100	17.348	97%

of computing all pairwise distances for GLS. However, this is observed in a setup where we use a straight-forward implementation of the k-means algorithm (without any efforts for optimization) and the parameters k and N are set to rather close values, i.e., 20 and 100, respectively. When we employ LC for the preprocessing stage, the clustering overhead is significantly reduced, but in return for some reduction in the effectiveness (cf. Table 3.2). In practice, some search engines may even prefer to pre-categorize the documents in its collection according to a taxonomy (as suggested in [AGHI09]) for various purposes (like improving the result relevance), and in this latter case the preprocessing cost of clustering can be totally avoided.

Nevertheless, even when the preprocessing times are included, C-GLS (C^2-GLS) with the k-means preprocessing yields an *overall* efficiency improvement of 78% (83%), 79% (83%) and 74% (78%) over GLS for TREC 2009, 2010 and Q1000 topic sets, respectively. Remarkably, the overall processing time for the C-GLS (C^2-GLS) algorithm drops under, respectively, 200 (150) milliseconds, which makes it possible to satisfy the demanding requirements of online query processing in real-life search systems. Furthermore, under heavy workloads, the search engines may even switch to a less effective yet more efficient preprocessing technique as a compromise, such as the LC method, which yields an overall processing time of less than 25 ms with

Table 3.6. Break-up of the diversification cost in terms of the key operation counts (per query).

Topic set	Algorithm	Preproc. algorithm	No. of rounds	No. of calls to $f(S)$	Time in $f(S)$	No. of iterations & look-ups *per call*	Time for look-ups
TREC 2009	GLS		54.220	8,288.100	741.108	1,600.0	97.936
	C-GLS	k-Means	10.740	1,703.380	34.460	400.0	4.820
	C^2-GLS	k-Means	3.388	23.320	0.473	400.0	0.015
	C-GLS	LC	10.964	1,743.184	35.669	400.0	5.051
	C^2-GLS	LC	8,327	70,469	1.377	400.0	0.076
TREC 2010	GLS		41.400	8,308.020	765.259	1,600.0	118.217
	C-GLS	k-Means	10.313	1,679.300	34.189	400.0	7.371
	C^2-GLS	k-Means	3.958	23.813	0.468	400.0	0.047
	C-GLS	LC	11.313	1,704,854	34.658	400.0	6.784
	C^2-GLS	LC	7.813	66.479	1.647	400.0	0.290
Q1000	GLS		19.875	5,526.915	539.655	1,600.0	109.588
	C-GLS	k-Means	8.534	1,276.450	23.303	400.0	1.793
	C^2-GLS	k-Means	1.034	17.694	0.321	400.0	0.026
	C-GLS	LC	8.605	1,280.213	23.456	400.0	1.855
	C^2-GLS	LC	2.291	51.433	1.031	400.0	0.139

C^2-GLS (an improvement of 97% over GLS) for all three topic sets.

In Table 3.6, we report the counts of key operations to shed light on the efficiency gains provided by our methods. Note that, unlike the processing time, these operation counts are independent of the experimental architecture, and hence, allows a more general comparison among the algorithms. Table 3.6 shows that the proposed approaches significantly reduce the average number of calls for computing the objective function and number of rounds till convergence, per query. Furthermore, the number of iterations within the objective function per call (which is also equal to the number of distance look-ups) is also reduced: as also illustrated in Figure 3.1, while C-GLS and C^2-GLS make only 400 iterations (and look-ups) per call, GLS requires 1600 iterations.

We also investigate the relationship between the operation counts and diversification time. It turns out that the majority of the diversification time is spent for computing the objective function (cf. Table 3.5). Therefore, the reductions provided by our methods in two ways, namely, in the number of calls for the objective function and number of iterations within the function, are almost exactly reflected to the diversification time. For instance, for the TREC 2009 set, GLS calls the objective function 8,288 times, and in each call of the function, the loop iterates 1600 times (cf. Algorithm 2); while for C-GLS (with k-means), these numbers are 1,703 and 400, respectively. Thus, in terms of the operation counts, C-GLS should be 19.5 times faster than GLS; which is actually reflected to the diversification times of 34.460 ms and 741.108 ms, respectively (i.e., implying a speed-up of 21.5 times). A similar proportionality is also observed between C-GLS and C^2-GLS: both methods iterate the same number of times (namely, 400) in the objective function; however, the former calls it 1,703 times whereas the latter calls only 23 times (again for the TREC 2009 dataset), a reduction of almost 74 times, which is almost perfectly reflected to the diversification times (i.e., 34.460 ms. vs. 0.473 ms, indicating a sped-up of 73.2 times).

Finally note that, the distance look-ups during the computation of the objective function take a non-negligible portion of the online diversification time (i.e., up to 20% for different algorithms and datasets). Our methods again significantly reduce the time for the look-ups, as shown in Table 3.6. While we cache all the distances during the preprocessing for all the methods compared here, for the scenarios where such a caching may not be possible and the look-ups have to be replaced by the actual distance computations, our reductions for the online diversification time would be more emphasized. Overall, all these findings confirm that the reductions shown in the algorithmic complexities (cf. Table 3.1) for the proposed methods are also reflected in the actual performance.

3.5.3 Evaluation of the Distributed Strategies

In this section, we investigate the impact of a distributed query processing architecture on the diversification effectiveness and efficiency of the implicit (i.e., GLS, C-GLS and C^2-GLS with the k-means) and explicit (xQuAD) diversification approaches, in a setup that employs either broker-based (BB-Div) or the node-based (NB-Div) diversification, as proposed in Section 3.4. To this end, we assume a simulated search cluster with $P = 10$ nodes. Given that the total collection is around 50 million documents, we believe the choice of cluster size is realistic, as each node is typically expected to include a few million documents (see, for instance [OAC$^+$12, CVK$^+$10] and the industrial practice[6]). In our simulation runs, we first retrieve top-1000 documents for a given query q, and randomly distribute[7] these results to each node so that each node stores 100 documents (i.e., $N = 100$). We repeated each simulation run five times and report the average results. As in the previous section, k is set to 20.

For broker-based diversification, each node returns its local top-20 results based on the relevance scores, resulting in up to 200 documents. From these documents, the broker selects the most relevant 100 document (as we keep $N = 100$ through all experiments for the sake of comparability) and executes the diversification algorithm on the top-100 set to create the final result set of 20 documents.

In the case of node-based diversification, each node applies the diversification algorithm to determine its local diversified result set of size 20 from its own 100 candidates. These diversified sets are merged at the broker and the global top-20 results are returned.

[6] http://www.searchtechnologies.com/enterprise-search-scalability.html

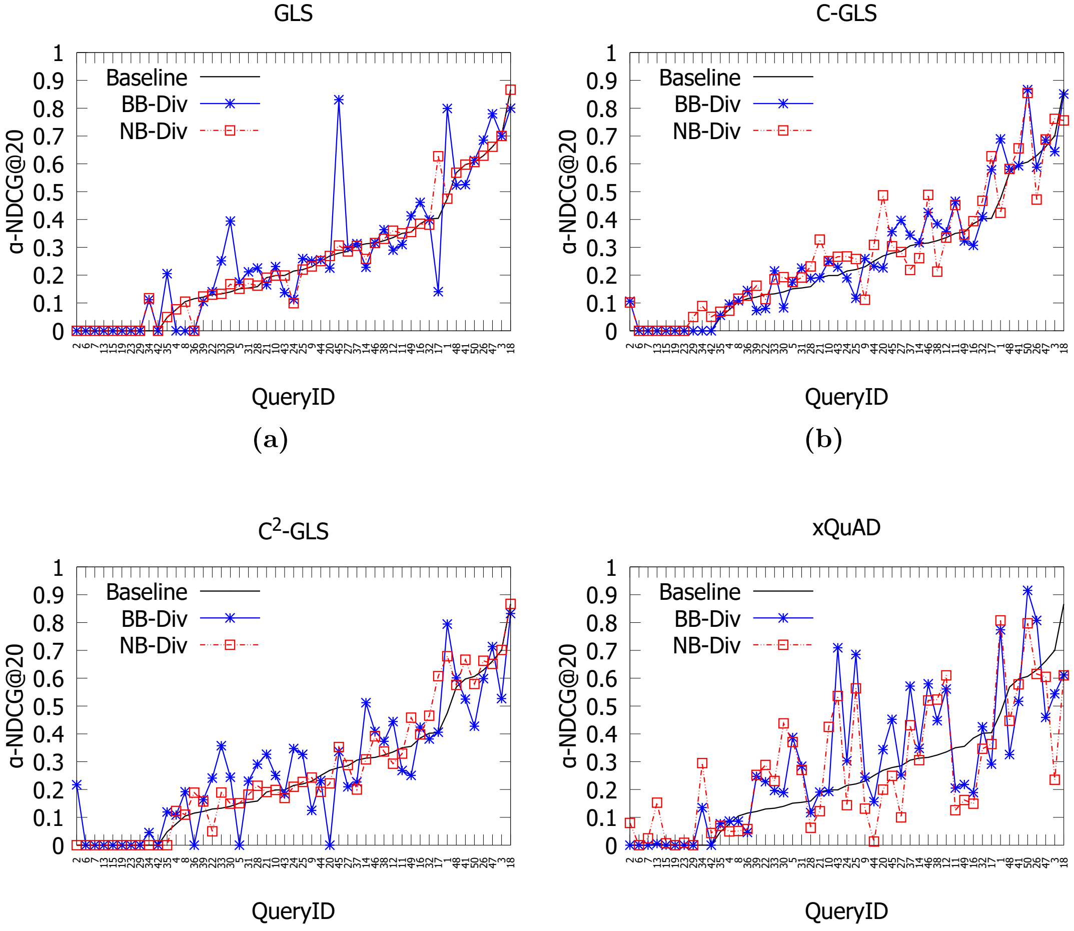
[7] We discuss alternative document partitioning strategies later in the subsection entitled *Critical Assumptions*.

Table 3.7. Retrieval effectiveness of distributed diversification algorithms for TREC 2009 and 2010 topic sets. The cases where the result of the BB-Div strategy differs significantly (at 0.05 level) from that of the NB-Div strategy are denoted with ‡.

Topic Set	Algorithm	Dist. Strategy	ERR-IA			α-NDCG			S-recall		
			@5	@10	@20	@5	@10	@20	@5	@10	@20
TREC'09	GLS	BB-Div	0.152	0.164	0.169	0.228	0.249	0.270	0.285	0.348	0.393
		NB-Div	0.124	0.138	0.147	0.187	0.219	0.254	0.257	0.351	0.423
	C-GLS	BB-Div	0.155	0.164	0.171	0.233	0.248	0.272	0.316	0.353	0.420
		NB-Div	0.142	0.159	0.167	0.214	0.248	0.281	0.280	0.361	0.459
	C^2-GLS	BB-Div	0.153	0.163	0.169	0.230	0.247	0.270	0.313	0.381	0.435
		NB-Div	0.133	0.147	0.156	0.195	0.227	0.265	0.258	0.345	0.421
	xQuAD	BB-Div	0.152‡	0.169‡	0.178‡	0.220‡	0.256‡	0.294‡	0.299‡	0.395‡	0.461
		NB-Div	0.136	0.149	0.160	0.202	0.229	0.275	0.267	0.347	0.465
TREC'10	GLS	BB-Div	0.138	0.156	0.166	0.184	0.221	0.255	0.276	0.386	0.483
		NB-Div	0.145	0.162	0.171	0.188	0.223	0.257	0.289	0.394	0.485
	C-GLS	BB-Div	0.168	0.187	0.195	0.209	0.249	0.278	0.287	0.408	0.507
		NB-Div	0.165	0.177	0.188	0.214	0.238	0.277	0.324	0.392	0.513
	C^2-GLS	BB-Div	0.164	0.175	0.184	0.210	0.231	0.265	0.310	0.385	0.506
		NB-Div	0.153	0.168	0.178	0.193	0.228	0.264	0.281	0.386	0.482
	xQuAD	BB-Div	0.177‡	0.194‡	0.200‡	0.234‡	0.272‡	0.293	0.390‡	0.487	0.526‡
		NB-Div	0.154	0.176	0.191	0.204	0.256	0.312	0.319	0.469	0.631

Effectiveness Evaluation. In Table 3.7, we provide the evaluation results using TREC 2009 and 2010 topics for all four algorithms combined with each of the two distributed diversification strategies. We see that broker-based and node-based diversification strategies exhibit a similar effectiveness, but the former is slightly better for the majority of algorithms and evaluation metrics. The differences between two distributed strategies are more visible for GLS (on TREC 2009 topics) and for xQuAD (on both topic sets), and found to be statistically significant (at 0.05 level using one-way ANOVA) for the latter algorithm.

In addition to average values, for each diversification strategy, we also provide a query-wise break up of performances in Figure 3.3 for each of the algorithms, namely, GLS, C-GLS, C^2-GLS, and xQuAD, respectively. In the plots, queries are sorted in the order of increasing α-NDCG@20 score obtained for the non-diversified baseline. We see that, in line with the general trends, diversification algorithms are usually outperforming the baseline. Interestingly, regardless of the layer of diversification (i.e., either at the broker or nodes), both xQuAD and GLS exhibit a more volatile behavior in that they improve certain queries a lot, but also hurting some others a lot. On the other hand, our algorithms (especially C-GLS) behave more conservatively, but while they usually improve the result diversity, the gains can be rather small for



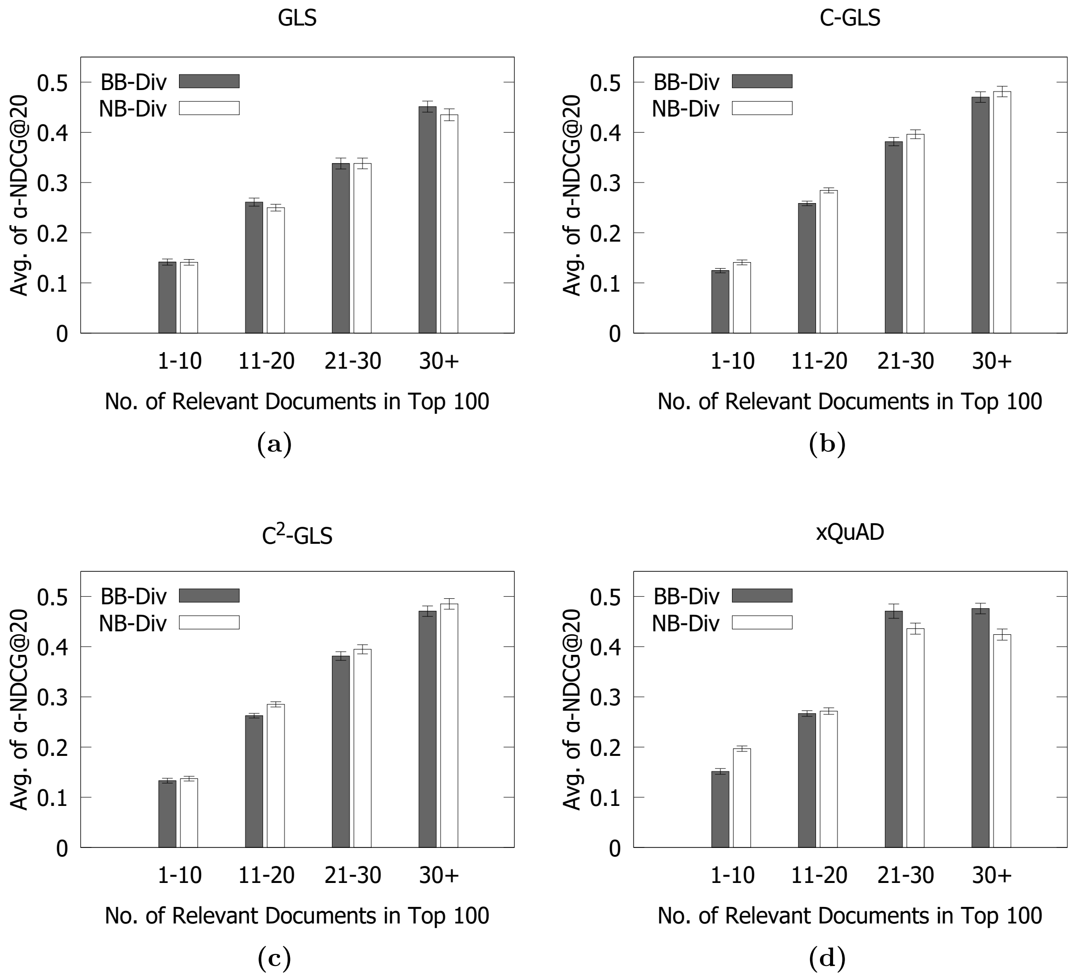
(a) **(b)**

(c) **(d)**

Figure 3.3. Query-wise α-NDCG@20 scores for BB-Div and NB-Div using GLS, C-GLS, C^2-GLS and xQuAD diversification algorithms (query ids sorted in ascending order of $\alpha-$NDCG@20 scores for the baseline).

many queries. Nevertheless, this is a positive finding for our cluster-based approaches as their diversification performance seem to be more robust over a set of queries.

Next, we concentrate on the possible causes of the performance differences between the distributed diversification strategies, especially for the GLS and xQuAD algorithms. For both of the latter algorithms, BB-Div seems to outperform NB-Div for the queries with higher α-NDCG scores (see corresponding plots in Figure 3.3). For a better insight, in Figure 3.4 (a) to (d), we show the effectiveness with respect to the number of relevant documents (based on the relevance judgments) encountered in the broker's candidate set D (i.e., top-100 results) for each query. The figure shows that, especially for xQuAD and GLS, the gap between BB-Div and NB-Div widens as the number of relevant documents in top-100 increases. This implies that, while



(a)

(b)

(c)

(d)

Figure 3.4. Effectiveness of distributed diversification strategies vs. the number of relevant documents in the top-100 results.

the NB-Div strategy works on a deeper pool (as each node operates on a different set of candidate documents), its pool may not include as many relevant documents as that of the broker. Supporting this latter hypothesis, for each query, Figure 3.5 shows the percentage of relevant documents in top-100 and top-1000 results, i.e., the broker's candidate set and the union of the nodes' candidate sets, respectively (as we distribute top-1000 results of a query among 10 nodes in the simulation runs). We see that while for some queries the percentage of relevant documents reaches up to 75% in top-100, the percentage of relevant documents in top-1000 does not increase proportionally, and indeed, it remains mostly less than 10%. In other words, the biggest possible advantage of NB-Div, being able to consider a deeper pool of documents, does not necessarily help in this setup, as the majority of the documents in the candidate sets of the nodes are indeed irrelevant.

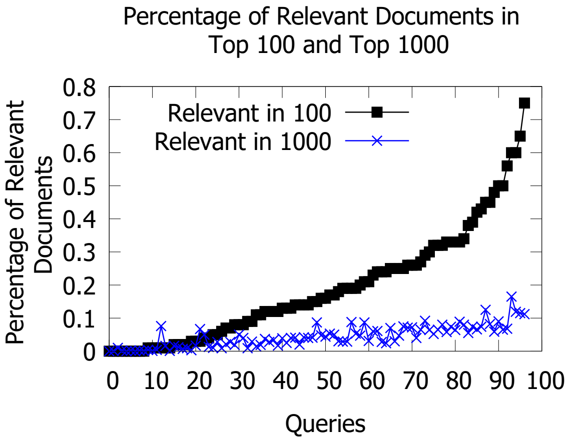


Figure 3.5. Percentage of documents judged as relevant in top-100 and top-1000 results for TREC 2009 and 2010 queries. Note that, x-axis represents the queries sorted wrt. the number of relevant documents in top-100 for a more clear visualization.

In this sense, the potential of the NB-Div strategy should not be underestimated: although it encounters many more irrelevant documents than the BB-Div strategy in the current TREC evaluation setting, it can still provide comparable results to the BB-Div strategy. We presume that in a setup where each query has more relevant documents and the initial retrieval strategy can retrieve more relevant documents in top-1000, NB-Div might outperform BB-Div. However, further investigation would require a test collection with a much higher number of relevance judgments. We identify the latter point as a limitation of the experimental framework provided by TREC Diversification Task: majority of the queries have a small number of relevant documents (i.e., on the average, there are around 200 relevant documents per query in TREC topic sets released between 2009 and 2012) and they are distributed quite unevenly among the sub-topics.

Efficiency Evaluation. In this section, we compare the efficiency of BB-Div and NB-Div in terms of the network communication costs. As a particular diversification algorithm is always executed on a candidate set of the same size either at the broker or at the nodes (and since the latter execution takes place in parallel), the processing cost of the algorithm itself (as extensively discussed in Section 5.2) would be the same for BB-Div and NB-Div. For this reason, our discussion in this section focuses only on the network communication costs. In our analysis, as in the previous section, we employ GLS (and its variants) and xQuAD as the representative methods for the implicit and explicit diversification approaches, respectively. However, the cost formulas developed here is applicable to any implicit diversification method that needs to access the actual document vectors of the candidate documents (e.g., our C-GLS and C^2-GLS, MMR [CG98b], etc.), and any explicit method that needs to compute the scores

Table 3.8. Parameters for the network cost computations.

Parameter	Description
$\lvert d\rvert$	no. of distinct terms in a document
s	size of an entry in the document vector (in bytes)
$\lvert A\rvert$	no. of aspects for a query
$\lvert a\rvert$	size of a query aspect (in bytes)
f	size of an entry in the vector of aspect scores (in bytes)
O_j	set of documents hosted at the node j
$\lvert V\rvert$	no. of nodes that include at least one candidate document
T	transfer rate of the network (MB/s)

of the candidate documents for each (known) query aspect (e.g, PM2 [DC12] and aggregation based methods in [OA15]). In this sense, our analysis sheds light on the general efficiency figures of the typical implicit and explicit diversification approaches on a distributed setup. Therefore, in the following cost formulas and experimental results, we prefer to label the cases as "implicit" or "explicit" diversification, rather than using particular algorithm names.

For the more straightforward NB-Div strategy, there is no additional network communication, assuming that each node stores the document vectors for the set of documents that are assigned to it, as well as the index on these documents[8], which is a practical assumption also employed in earlier works (e.g., [OAC+12]). In contrast, the BB-Div strategy requires that the necessary information for the diversification stage, which naturally depends on the type of the utilized algorithm, should be transferred to the broker. In what follows, we analyze this latter case in detail.

In Table 3.8, we list the parameters and their symbols to be used in the cost formulas, and in Table 3.9, we provide the formulas for the network cost of the implicit and explicit algorithms when employed together with the BB-Div strategy. Note that, the communication volume is computed by summing the total amount of data (in bytes) that needs to be transferred on the network. In contrast, network communication time is computed as the time to transfer a data package to the broker from the node that sends the *maximum* amount of data (Note that, the number of candidate documents stored in a node and their total size may differ among the nodes).

We derive the formulas in Table 3.9 based on the following facts. If BB-Div employs an implicit diversification algorithm, it will need to fetch the document vectors of the candidate results (so that the pairwise document similarities, as in GLS or MMR [CG98b], or document-cluster similarities, as in the case of our C-GLS and C^2-GLS methods, can be computed). Hence, the network communication volume is simply the sum of the lengths of the document vectors for all candidate documents in

[8] We discuss alternative storage strategies for document vectors later in the subsection entitled *Critical Assumptions*.

Table 3.9. Network communication costs for the implicit and explicit diversification approaches with BB-Div strategy.

Algorithm type	Communication volume	Communication time
Implicit	$\sum_{d_i \in D} \left\| d_i \right\| \times s$	$\max_{j \in \{1,\cdots,P\}} \frac{\sum_{d_i \in D \cap O_j} \|d_i\| \times s}{T}$
Explicit	$\left(\sum_{a_i \in A} \|a_i\| \times \|V\| \right) + \left(N \times \|A\| \times \|f\| \right)$	$\max_{j \in \{1,\cdots,P\}} \frac{\sum_{a_i \in A} \|a_i\| + \sum_{d_i \in D \cap O_j} (\|A\| \times f)}{T}$

D. In contrast, for the communication time, we compute the transfer time from each node to the broker, which is based on the total document length of the candidate documents hosted at a node, and take the *maximum* of these transfer times, as the broker needs to wait until all data from the nodes arrive.

In case of an explicit algorithm, such as xQuAD, the broker should compute the score of each query aspect for each candidate document using a retrieval model. This latter score can be computed by either transferring the document vectors to the broker, of which cost already discussed for the BB-Div case, or more conveniently, sending the query aspects to those nodes that host the documents in the candidate set. In the communication volume formula for the explicit case in Table 3.9, the first summation represents the size of the query aspect strings (in bytes), which is the amount of data that is sent to each node including a document in D. Then, each such node computes the aspect-document score by using its local inverted index, and sends back to the broker a vector that involves the score of each candidate document for each aspect. The overall communication volume incurred by this latter stage is $N \times |A| \times f$, where A denotes the set of query aspects, f denotes the size of a score value, and N is the size of the candidate set, as before.

To apply the cost formulas in Table 3.9 in our experimental setup, we set the parameter s as 8 bytes; assuming that each entry in the document vector will include two integer values, a term id and its frequency in the document. We also set the parameter f as again 8 bytes, assuming that the score of an aspect for a document is stored as a double value. The transfer rate of the network, assuming a LAN, is set to 11 MB/s. Note that, we discuss other possible values for these parameters in the following subsection.

Table 3.10 reveals that, as expected, explicit diversification algorithms incur network costs that are two orders of magnitude smaller than those incurred by the implicit algorithms; and hence, if query aspects are available beforehand, employing an explicit algorithm on top of the BB-Div strategy is more efficient. In contrast, for the practical scenarios where no aspects are known and implicit methods need to applied, BB-Div strategy incurs some overheads in terms of the communication volume

Table 3.10. Network cost in terms of the communication volume (in bytes) and time (in milliseconds) per query for the BB-Div strategy.

Topic set	Div. Type	Comm. volume	Comm. time
TREC'09	Implicit	391,360	5.159
TREC'10	Implicit	433,658	5.749
TREC'09	Explicit	5,455	0.066
TREC'10	Explicit	4,845	0.058

and time. However, we envision that these additional costs are still affordable. For instance, in a real-life setting, a typical query of 15 characters on the average (e.g., see [KB06]) need to be send to several thousands of nodes, say, 50,000, at a particular data centre (This is a moderate estimation given that Microsoft had more than 1 million servers in 2013[9]). With a back-of-the-envelope computation, we see that even forwarding a query string to the nodes in the latter setup causes a communication volume of 750,000 bytes, which is larger than the data volumes shown in Table 3.10 (e.g., 391,360 and 433,658 bytes per query for TREC 2009 and 2010 sets, respectively) . Furthermore, the communication volume formula in Table 3.9 only involves N, the size of the candidate set, as a parameter, but not the number of nodes; and hence the overhead will be the same regardless of the number of nodes for a fixed N. Regarding the network communication time, the cost is around 5 ms using a moderate parameter for the network transfer rate (i.e., 11 MB/s); and should be clearly affordable in a real life setup. Therefore, we conclude that, when the candidate set D is small, the network costs seem to be an affordable overhead; and BB-Div remains as a viable option for applying implicit diversification in a distributed setup.

Critical Assumptions. In our evaluation of the diversification algorithms in a distributed setup, we have some critical assumptions that are essentially based on the common practice for Web search setup and may not hold in different scenarios. We list and discuss these assumptions as follows:

- Distribution of the collection: As discussed in the related work section, it is usually assumed that the state-of-the-art method used in partitioning the collection of a search engine is document-partitioning, where each node (and its possible replicas) in the system is responsible for a disjoint subset of the collection and the corresponding index. In the literature, different approaches for assigning documents to the nodes are proposed. The most straightforward approach, as we also assume here, is a random allocation. In contrast, alternative partitioning approaches usually aim to store the similar documents at the same node, which can be achieved via unsupervised clustering, using semantic catalogues or exploiting previous query results in the log (e.g., [PSL06, CBY11]). Despite its simplicity, the random document-partitioning is attractive in many ways: first,

[9] http://www.datacenterknowledge.com/archives/2013/07/15/ballmer-microsoft-has-1-million-servers/

its implementation is practical, as a hash function can be used to quickly assign each document to a node. In contrast, alternative approaches require running an algorithm to determine the document's node. Second, random partitioning achieves good load balancing. In contrast, alternative document partitioning models usually suffer from the load imbalance, i.e., some nodes need to answer a large number of queries while some others stay idle. Finally, in the random partitioning, fault tolerance may be simply achieved by having a fixed number of replicas for each node. For the alternative approaches, this issue is also more complicated due to the load imbalance problem; i.e., the nodes that include the documents from the most-popular sites maybe accessed more, and need to have a larger number of replicas. In the light of these discussions, we believe that the random document-partitioning is the most practical approach and hence, employed in the large-scale search engines, as also stated in [FLSV11]. Nevertheless, for the scenarios where topically similar documents are assigned together to a single node (e.g., a meta-search scenario as in [KC10]), the query processing model would also change (i.e., include a resource selection stage instead of forwarding the query to all the nodes) and hence, our findings discussed in the previous sections may not hold. Clearly, such scenarios are not in the scope of our work, and can be investigated in the future work.

- Index and document servers: In this work, following the practice in the earlier studies [OAC$^+$12], we assume that a particular node stores both a disjoint subset of the collection, and its index. It is also possible that the document subset and its corresponding index are stored in physically different servers; i.e., at a document and index server, respectively (e.g., see [Dea09]). In this case, implicit diversification methods with NB-Div strategy would need to fetch the document vectors from another server, yielding network costs similar to those in BB-Div case. For the explicit diversification methods with NB-Div strategy, there would be still no network costs, as these approaches use the index to compute the score of each candidate document for each query aspect.

- System parameters: In our cost formulas shown in Table 3.9, we set the size of a document vector entry as 8 bytes, assuming that such a vector is composed of integer pairs that represent a term identifier and its frequency in the document. In practice, such a vector can have additional information and/or can be stored in a more efficient way, i.e., in a compressed form. We also assume a network speed of 11 MB/s, which might be very moderate for connecting the servers in a data center. Nevertheless, we believe that replacing such parameters with more realistic values will not change the trends reported in this chapter.

3.6 Summary of the key findings

Our key findings in this chapter can be summarized as follows:

- Explicit diversification approaches are both effective and efficient; as xQuAD, a representative explicit approach, spends around only 10 ms for diversification; whereas the implicit algorithms are either efficient yet less effective (as in the case of LCD) or effective yet inefficient (as in the case of the original GLS). Given that the explicit aspects of a query may not be always available in practical scenarios, this finding also justifies our first goal in this chapter, namely, improving the efficiency of GLS as a promising implicit algorithm.

- Using clustering as a basis of the diversification on its own does not yield high quality results, as LCD is found to be only slightly more effective than the non-diversified baseline. In contrast, our methods C-GLS and C^2-GLS that employ the clustering as a preprocessing stage for GLS are found to be both effective and efficient. In particular, when the k-means algorithm is used for the preprocessing, their effectiveness is comparable to (or, sometimes better than) GLS, whereas the overall diversification time is reduced by more than 80%. It is also possible to improve the diversification efficiency by employing a cheaper preprocessing algorithm, namely LC; that yields slightly inferior diversification quality in return to higher efficiency. For this latter case, C^2-GLS takes at most 23 ms, which means a 97% improvement over GLS. These findings mean that the proposed algorithms can be utilized in real-world scenarios with strict budgets for query processing.

- Our experiments on a distributed setup show that running the diversification algorithms (of either implicit or explicit type) at the broker (i.e., using BB-Div) yields higher effectiveness scores than applying diversification at each node (i.e., using NB-Div). Our detailed analysis reveals that the ineffectiveness of NB-Div might be caused by the relatively small number of relevant documents per query in TREC datasets. Because of this, the candidate sets at the nodes include a larger number of irrelevant documents, and hence, lead to inferior diversification effectiveness.

- We also show that NB-Div is relatively cheap, as it incurs no network communication overhead in a typical distributed setup. In contrast, BB-Div has additional overhead in terms of the network costs (especially for the implicit diversification algorithms); however these costs, namely, network communication volume and time, seem to be affordable in a practical Web search setup.

In the light of above findings, we can claim that in a setup that needs an implicit diversification algorithm, the proposed methods C-GLS and C^2-GLS (with the k-means or LC preprocessing) can be safely utilized as effective and efficient variants of GLS. Furthermore, if a given query is expected to return relatively small number of relevant documents, it may be better to apply these algorithms at the broker (as the network communication overhead seems to be affordable in a realistic setup); otherwise, applying the diversification at the nodes would be a more efficient choice.

3.7 Conclusion

For practical application of diversification in a large-scale setting, two requirements need to be met. First, we need an algorithm with low computational complexity to satisfy the demanding efficiency requirements of online query processing. Second, the diversification process should be executable on a computing cluster where each node holds a collection partition, because larger collections cannot be maintained on one central node.

In this chapter, we presented C-GLS and C^2-GLS, two greedy algorithms that perform an initial document clustering to reduce the GLS complexity from quadratic to linear (with the number of candidate documents). We show that the proposed approaches can reduce the online diversification cost by more than 80% and up to 97%, while achieving comparable or even better effectiveness than the GLS solution.

We also studied how distribution of the diversification process affects its result quality and efficiency. In our experiments, diversification on the broker with the BB-Div strategy yielded better result quality than diversification on the nodes with the NB-Div strategy; however there were also cases where both strategies performed equally well. While evaluating their efficiency, we found that the diversification algorithms with BB-Div strategy incur additional costs for the network communication (while NB-Div incurs no network costs); fortunately, this seems to be an affordable overhead in real-life settings.

These two contributions pave the way for scalable distributed diversification of search results for Web-scale document collections. We also anticipate that our work may lead to the community interest towards the development and evaluation of diversification algorithms on distributed architectures, which we believe to be the next and natural test-bed for evolving research in this field.

In our future work, we plan to evaluate the distributed diversification for other scenarios that employ alternative document allocation policies. We also plan to investigate approaches to further reduce the network communication costs when diversification is applied at the broker.

4 Exploiting Result Diversification Methods for Feature Selection in Learning-to-Rank

As discussed in Chapter 2.2, feature selection approaches try to identify a subset of features for improving the effectiveness and efficiency of machine learning methods. In this Chapter, we exploit different diversification algorithms for improving feature selection in learning-to-rank (LETOR). The motivation is that diversification algorithms can be adopted to estimate a trade-off between the relevance and diversity of information provided by different types of features.

4.1 Introduction

Learning-to-rank (LETOR) is the state of the art method employed by the large-scale commercial search engines to rank the search results. Given the large number of features available in a search engine, which is in the order of several hundreds (e.g., see Yahoo! LETOR Challenge [CC11]), it is desirable to identify a subset of features that yield a comparable effectiveness to using all the features. Since search engines typically employ a two-stage retrieval where an initial set of candidate documents are re-ranked using a sophisticated LETOR model, a smaller number of features would reduce the feature computation time, which must be done on-the-fly for the query dependent features, and hence overall query processing time [DBC13]. Furthermore, improving the efficiency of the LETOR stage would allow retrieving larger candidate sets and, subsequently, can help enhancing the quality of the search results.

In a recent study, Geng et al. proposed a filtering-based feature selection method that aims to select a subset of features that are both effective and dissimilar to each other [GLQL07]. Inspired from this study, we draw an analogy between the feature selection and result diversification problems. In the literature, a rich set of greedy diversification methods are proposed to select both relevant and diverse top-k results for web search queries (e.g., see [CG98b, GS09, WZ09b, RBS10, SCAC13]). We apply three representative diversification methods, namely, Maximal Marginal Relevance (MMR) [CG98b], MaxSum Dispersion (MSD) [GS09] and Modern Portfolio Theory (MPT) [WZ09b, RBS10] to the feature selection problem for LETOR. To the best of our knowledge, none of these methods are employed in the context of learning-to-rank with the standard search engine datasets.

In the next section, we first describe the baseline strategies for the feature selection from the literature, and then discuss how we adopt the result diversification methods for this purpose. In Sections 4.3 and 4.4, we present the experimental setup and evaluation results, respectively. Finally, we conclude in Section 4.5.

4.2 Feature Selection for LETOR

Feature selection techniques for the classification tasks are heavily investigated in the literature and fall into three different categories, namely, filter, wrapper and embedded approaches [GLQL07]. Strategies in the filter category essentially work independently from the classifiers and choose the most promising features in a preprocessing step. In contrast, the strategies following the wrapper approach consider the metric that will be optimized by the classifier whereas those in the embedding category incorporate the feature selection into the learning process. Earlier studies also show that such feature selection methods do not only help improving the accuracy and efficiency of the classifiers, but may also introduce diversity in ensembles of classifiers [CC00].

For learning-to-rank, there are only a few recent studies that address the feature selection issue [GLQL07, DC10]. Following the practice in [GLQL07], we focus on the feature selection methods that fall into the filter category.

4.2.1 Preliminaries

For a given feature $f_i \in F$, we obtain its relevance score for a query by ranking the results of a query solely on this feature and computing the effectiveness for the top-10 results. The effectiveness can be measured using any well-known evaluation measure (like MAP, NDCG) or a loss function (as in [GLQL07]). In this study, we employ NDCG@10 as the effectiveness measure and denote the *average* relevance score of a feature over all queries by $rel(f_i)$. To capture the similarity of any two features, denoted with $sim(f_i, f_j)$, we compute the Kendall's Tau distance between their top-10 rankings averaged over all queries (as in [GLQL07]). The objective is selecting a subset of k features (F_k), where $k < |F|$, such that both the relevance and diversity (dissimilarity) among the selected features are maximized.

4.2.2 Baseline Feature Selection Methods

Top-k Relevant (TopK): A straightforward method for feature selection is choosing the top-k features that individually yield the highest average relevance scores over the queries [DC10].

Greedy Search Algorithm (GAS): This is the greedy strategy proposed by Geng et al. in [GLQL07]. It starts with choosing the feature, say f_i, with the highest average relevance score into the set F_k. Next, for each of the remaining features f_j, its relevance score is updated with respect to the following equation:

$$rel(f_j) = rel(f_j) - sim(f_i, f_j) \cdot 2c, \tag{4.1}$$

where c is a parameter to balance the relevance and diversity optimization objectives. The algorithm proceeds in a greedy manner by choosing the next feature with the highest score and updating the remaining scores, until k features are determined.

4.2.3 Diversification Methods for Feature Selection

As the astute reader would realize, the goal of feature selection as defined in [GLQL07, DC10] is identical to that of the search result diversification techniques: both problems require selecting the most relevant and, at the same time, diverse items. Motivated by this observation, we adopt three different implicit result diversification techniques to the feature selection problem, as follows.

Maximal Marginal Relevance (MMR): This is a well-known greedy strategy originally proposed in [CG98b]. Peng et al. propose a similar idea of minimal-redundancy maximal-relevance in [PLD05]. In a recent study [CORA], MMR is employed for feature selection in learning-to-rank in a setup with a limited number of social features, but not evaluated on the standard search datasets, as we do in this chapter.

In this study, we adopt a version of MMR described in [VRB+11]. The MMR strategy also starts with choosing the feature f_i with the highest relevance score into the F_k. At each iteration, MMR computes the score of an unselected feature f_j according to the following equation:

$$mmr(f_j) = (1-\lambda)rel(f_j) + \frac{\lambda}{|F_k|} \sum_{f_i \in F_k} 1 - sim(f_i, f_j), \lambda \in [0,1], \tag{4.2}$$

where λ is again a trade-off parameter to balance the relevance and diversity.

MaxSum Dispersion (MSD): An alternative representation of the diversification (and hence, feature selection) problem is casting it to the facility dispersion problem in the operations research field [GS09]. In this case, our objective in this chapter, i.e., maximizing the sum of relevance and dissimilarity in F_k, can be solved with the greedy 2-approximation algorithm that is originally proposed for the well-known MaxSum

Dispersion (MSD) problem. In the MSD solution, a pair of features that maximizes the following equation is selected into F_k at each iteration:

$$msd(f_i, fj) = (1 - \lambda)(rel(f_i) + rel(f_j)) + 2\lambda(1 - sim(f_i, f_j)), \tag{4.3}$$

where λ is the trade-off parameter.

Modern Portfolio Theory (MPT): This approach is based on the famous financial theory which states that one should diversify her portfolio by maximizing the expected return (i.e, mean) and minimizing the involved risk (i.e., variance). In case of the result diversification, this statement implies that we have to select the documents that maximize the relevance and have a low variance of relevance [WZ09b, RBS10]. The latter component has to be treated as a parameter and its best value can be computed by sweeping through the possible values (as in [WZ09b]) unless additional data, such as click logs, are available [RBS10].

Fortunately, in case of the feature selection for LETOR, we have adequate data to model both the mean and variance of the relevance of a feature. Obviously, mean relevance of a feature is $rel(f_i)$ as we have already defined. For the variance of a feature ($\sigma^2(f_i)$), we compute the relevance score of f_i for each query q, and then compute the variance for this set of scores in a straightforward manner. Thus, the greedy MPT solution chooses the feature that maximizes the following equation at each iteration:

$$mpt(f_i) = rel(f_i) - [b\sigma^2(rel(f_i)) + 2b\sigma(rel(f_i)) \sum_{f_j \in F_k} \sigma(rel(f_j)) * sim(f_i, f_j)]. \tag{4.4}$$

Note that, we eliminated the rank position component from the original formula [WZ09b, RBS10] as it does not make sense for the feature selection problem. As before, $b \in [0, 1]$ is the trade-off parameter to balance the relevance and diversity.

4.3 Experimental Setup

Datasets. Our experiments are conducted on three standard LETOR datasets, OHSUMED[1] from Letor3.0, MQ2008 from Letor4.0 and SET2[2] from Yahoo! LETOR Challenge. In Table 4.1 we summarize the characteristics of each dataset. The Yahoo! SET2 has 596 features and is also the largest dataset with respect to the number of queries and instances. But previous studies have also shown that even for a small number of features, feature selection can improve the ranking [GLQL07].

[1] http://research.microsoft.com/en-us/um/beijing/projects/letor

[2] http://webscope.sandbox.yahoo.com/catalog.php?datatype=c

Table 4.1. Datasets

Dataset	No. of queries	No. of annotated results	No. of features
OHSUMED	106	16,140	45
MQ2008	800	15,212	46
Yahoo! SET2	6,330	172,870	596

LETOR algorithm. Our evaluations employ RankSVM [HGO00], which is a very widely used pairwise LETOR algorithm. More specifically, we used SVMRank[3] library implementation. We trained the classifier with a linear kernel with $\epsilon = 0.001$. We report the results with the C values (where $C \in [0.00001, 10]$) that yields the best performance on the test set with all the features.

Evaluation measures. We evaluate all the feature selection methods using 5-fold cross validation for the OHSUMED and MQ2008 datasets. For Yahoo! SET2, we use the training and test sets as provided. The evaluation measures are MAP and NDCG@10.

4.4 Experimental Results

In Figure 4.1, we report the NDCG@10 and MAP scores obtained on the OHSUMED dataset using the baseline and proposed feature selection methods. We observe that when the number of selected features is greater than 10, the performance is comparable or better than using all features (ALL). Furthermore, the methods adapted from the diversity field outperform the baselines (TopK and GAS). In particular, MPT is the winner for both evaluation measures when the number of features is set to 15 or 20.

Figure 4.2 shows the performance for the MQ2008 dataset. In this case, the feature selection algorithms can reach the performance of the ALL only after selecting more than 15 features. For the majority of the cases, the methods adopted from the diversification field are again superior to the baselines, and MSD is the winner method for this dataset.

Finally, in Figure 4.3, we report the performance for the Yahoo! SET2. As the experiments take much larger time on this dataset, we only present the results for selecting 100 features (out of 596). We observe that, feature selection methods with 100 features cannot beat the all features baseline ALL (not shown in the plots), which is reasonable as we only use one sixth of the available features. MPT is again the best adapted method, and it outperforms TopK baseline for both evaluation measures, and better than or comparable to GAS for MAP and NDCG measures, respectively.

The statistical significance of our methods is verified using the paired t-test with

[3] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

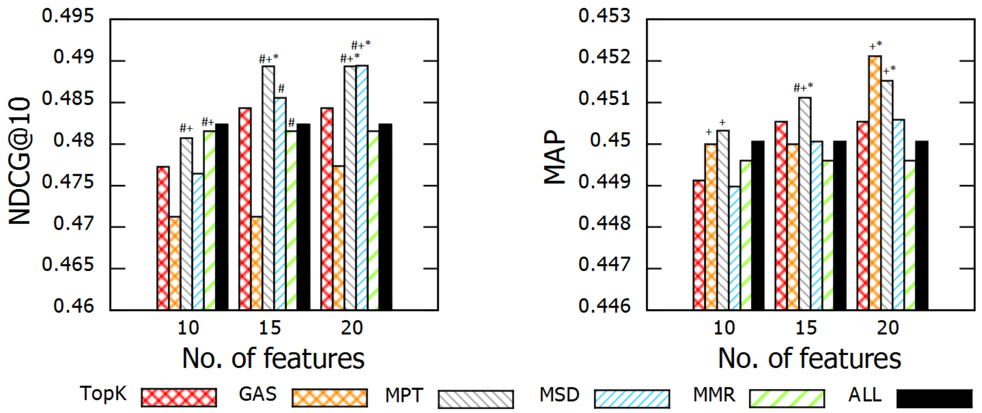


Figure 4.1. Ranking effectiveness on OHSUMED: NDCG@10 (left) and MAP (right).

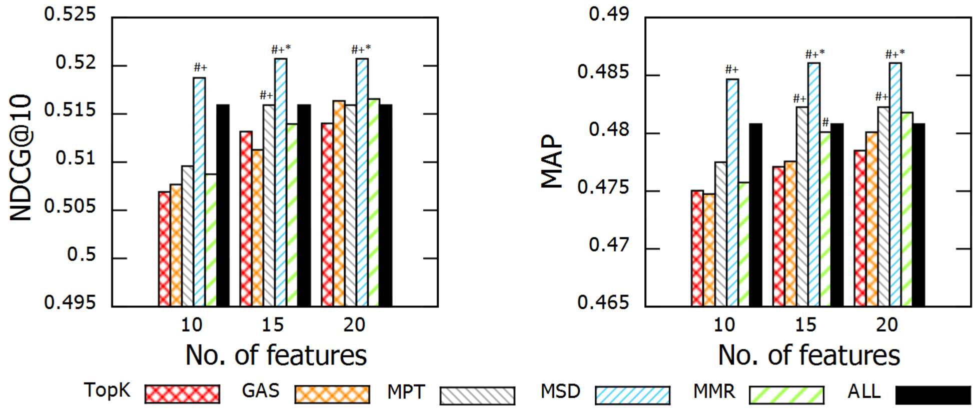


Figure 4.2. Ranking effectiveness on MQ2008: NDCG@10 (left) and MAP (right).

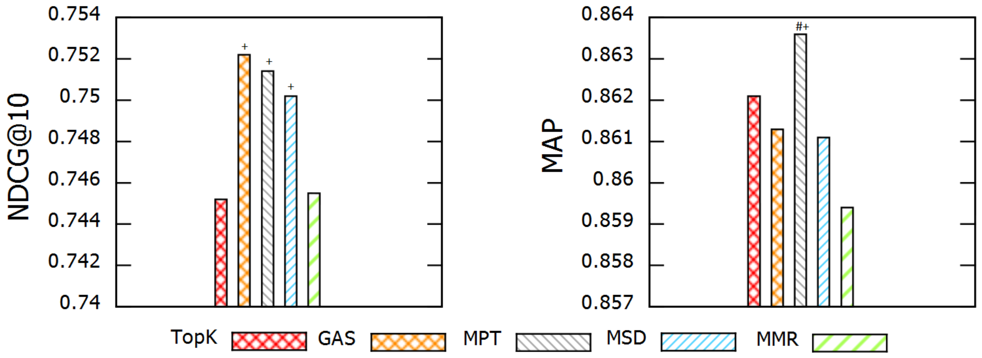


Figure 4.3. Ranking effectiveness on the Yahoo! SET2: NDCG@10 (left) and MAP (right).

$p < 0.05$. In Figures 1-3, we show the significant differences to the baselines TopK (denoted with +), GAS (denoted with #) and ALL (denoted with *).

4.5 Conclusions

In this chapter, we adopted several methods from the result diversification field to address the problem of feature selection for LETOR. Our evaluations showed that these methods yield higher effectiveness scores than the baseline feature selection strategies for various standard datasets.

5

Analyzing and Predicting Privacy Settings in the Social Web

Social networks provide a platform for people to connect and share information and moments of their lives. The increase amount of personal information on the web makes privacy a major issue for users and social networks. Due to carelessness, unawareness or difficulties in defining adequate privacy settings, private or sensitive information may be exposed to a wider audience. Although these causes usually receive public attention, e.g. when it involves celebrities, the general public is also subject to these issues. In this chapter, we envision a mechanism that can suggest users the appropriate privacy setting before sharing content on the web. The contributions in this Chapter are as follows. First, we present a through analysis on usage of privacy settings in the social web. Second, we introduce a classification approach for predicting highly private social content. Finally, we present a set of key feature-categories which can be used for predicting highly private social content.

5.1 Introduction

Social networking sites such as Google+, Twitter and Facebook allow their users to post updates, tweets, pictures, links and videos to their circles of friends, their followers or to the whole world. By doing so, users generate a digital footprint that defines their "online presence".

Similar to the "offline world", the various digital platforms provide means to define and structure a user's social network. Most of the services support a way to define different groups for information sharing within a user's social network, although each service provides its own implementation and terminology for this. Earlier social networks like Orkut had communities, for example. A member of a community could share posts, pictures and different sorts of information that were only visible to the members. In a similar fashion, current social network platforms such as Facebook and

Google+ provide analogous features with Facebook Groups and Google+ Circles. The common goal is to facilitate the social network users to manage the audience of their interactions and shared content.

The use of such structuring facilities becomes a necessity, because we increasingly share the same social networking sites with persons from different spheres of our real world social networks, such as colleagues, acquaintances, friends and family members. Since each of these groups represent different aspects of our lives, it is often desirable to also be able to maintain this separation in the digital world. For example, certain family affairs might better not be shared with colleagues or acquaintances. Maintaining the right balance of interaction and involvement within those social groups helps us to manage our different roles in life. For this purpose, social networks support their users to manage their groups and to keep control of their privacy settings. Unfortunately, in many cases, these settings are buried in menus, tabs and configurations that are notoriously hard to understand for the regular user [MJB12]. Contributing to this discussion, Facebook founder Mark Zuckerberg claimed that the rise of social networking online means that people no longer have an expectation of privacy, adding "we decided that these would be the social norms now and we just went for it" [Joh10].

As a consequence, in the past years, there have been several cases of people who involuntarily, unknowingly 'leaked' information to the wrong audience. Common cases include public messages that were supposed to be privately sent to one particular recipient, and posts that are targeting a specific audience and are in fact publicly available (a recurrent issue on Twitter). The presence of inadequate or inappropriate information about a person in the public sphere can have serious impact, for example on employment opportunities. Exemplary cases are reported in [MS09]. An indication for the increasing awareness for this topic is the current legal discussion about the *right to be forgotten* now called the *right to erasure* in the European community. This discussion addresses the right of individuals not to be stigmatized as a consequence of a specific action performed in the past [Man13]. Although there is a distinction between the right to be forgotten and the right for privacy - the right for privacy constitutes information that is not publicly known, whereas the right to be forgotten involves removing information that was publicly known - there is a clear link: if people unintentionally share information to wrong audiences, they might later regret it and want the information to be 'forgotten'. Ideally, it should be prevented that such information would be unintentionally publicly shared in the first place.

This implies that there is a need for better support for selecting adequate privacy settings in social networks. With that in mind, in this work, we investigate to what extent it is possible to predict the privacy setting of posts. We build our work on top of Facebook's privacy settings. Facebook is arguably the most popular social network and it provides its users a range of privacy options. In order to understand the users' privacy behavior, we first provide an analysis of privacy settings for Facebook posts. Subsequently, we present a method for predicting privacy settings by employing classification based on a small but effective set of features that are available at post creation time. Evaluations show that privacy settings can be predicted with high accuracy, which may allow automatic privacy-setting assistance for the end users and third party apps.

The remainder of this chapter is structured as follows. In Section 5.2, we summarize the relevant related work. In Section 5.3, we describe our efforts to collect the sensitive data, followed by a data analysis in Section 5.4. In Section 5.5, we describe the experiments and results towards a privacy prediction method. We finally discuss and conclude our work in Section 5.6.

5.2 Related work

Social media sites, such as Twitter, Facebook and Google+, are designed to share information - and other content, such as pictures, videos and links - with other users. Studies on the usage of social media platforms focus, among others, on usage motivations [Joi08], user behavior [LES08], and relations and social capital [ESL11, EGV+13]. A recent study on Facebook [ZSN+13] shows that the dynamic and temporal changes of the relationships between users lead to conflicting privacy needs of the user.

Apart from relatively harmless updates, such as sharing a link or other types of public content, messages on Twitter and Facebook may contain highly personal information such as the user's location or email address. For this reason, social media sites typically offer their users several ways for indicating the intended audience of shared messages. First of all, there are *default* settings, which can be adapted by the user. Second, users can overrule these default settings for specific messages. Third, in many cases it is possible to delete, hide or edit a message post hoc.

However, as indicated by several studies (e.g. [LGKM11]), users often do not inspect or adapt the default settings offered by the system; thus, most messages are sent with the default settings. Due to this behavior, messages often have a wider audience than intended or expected by the user. According to a recent report from the Pew Internet & American Life Project [Mad12], particularly males and young adults have posted content that they regret. Not surprisingly, these are also the users with the least restricted privacy settings. However, due to the raising awareness of privacy issues and their implications, more and more users actively manage their privacy settings and prune their profiles.

Other studies on Facebook privacy analyze user concerns regarding sharing personal content with a public audience or with third-party applications [GA05, FE08]. Similarly, in YouTube it has been observed [Lan07] that users follow different strategies for balancing the pros and cons of sharing with privacy. As an example, users do share videos with private content, but can ensure that their faces are not displayed and their identities are not disclosed. In the context of mobile apps, it is again reported that users' privacy settings are diverse, yet can be represented via a relatively small number of privacy profiles [LLS14, LLSH14].

Still, research has shown that users typically disclose more personal information online than they would do in face-to-face situations. There are many risks associated with content that is unknowingly disclosed to the public. Some of these risks - including mobbing, loss of reputation, family problems and lost career opportunities - are summarized in [Ros11, TWC12]. A remarkable initiative to raise attention for these issues is the site PleaseRobMe[1], which aggregates and shows tweets of users who report to be away from home. In addition, the user is informed via a (public) tweet.

With the goal of raising awareness for the problems related to sensitive information leaks and privacy settings, Kawase et al. [KNH+13] introduced FireMe!, a website that contains live streams of people who publicly tweet offensive comments towards their working environment, bosses and coworkers. In their work, they built a system that, once an offensive message was detected in the twittersphere, the author of the offending tweet was sent an alert message. Their results show that only 5% of the users who were alerted by the system later on deleted the compromising tweet. The authors called for the deployment of an alert system that prompts users before a compromising tweet is sent. In fact, our work goes into this direction. By understanding and predicting privacy settings, we might be able to advise (suggest) users the appropriate privacy settings for a given post, before it is effectively out there.

There are other works in the literature that aim to recommend privacy settings. Fang et al. suggest building models that can predict whether a user's friends should be allowed to see certain attributes (such as the birth date or relationship status) in the Facebook profile of the user [FKLT10, FL10]. Similarly, Ghazinour et al. build a classifier to predict the privacy-preference category of a user (such as "pragmatic" or "unconcerned"). Furthermore, they employ a simple kNN approach to determine the similar users to a given user, and based on the preferences of these similar users, they suggest privacy settings, again, for the attributes in the user's Facebook profile [GMS13]. Our work differs from those in that we do not address such general attributes but we aim to recommend a privacy setting for every post (be it a status update, a video link or a photo) made by the user. Machine learning and/or collaborative filtering methods are further employed to recommend privacy settings for the location-sharing services [TSH10, XKJ14] and mobile apps [LLSH14]. The latter domains involve different dynamics and/or features for setting privacy options than those in Facebook; the social network addressed in our study.

In our approach, we aim to directly support users in choosing privacy settings at the moment that they submit a post. This goal is similar to the work presented in [ZSHD12], although addressing a different media type. In their work, Zerr et. al propose a method for detecting private photos in Flickr that are posted publicly by

[1] http://pleaserobme.com/

extracting a set of visual features. Their results show that a combination of visual and textual features achieves a considerable performance for classifying and ranking private photos. While following a similar goal, we operate in a different setting: we use social network specific features and we aim to predict more fine-granular privacy settings. In our work in Chapter 7, we have also used different types of social network features, but for different prediction tasks, namely for suggesting Facebook posts for content retention and summarization.

5.3 Dataset

In this section we present the two datasets that we used for our experiments. Both datasets have been collected using an experimental Facebook App[2]. This app has been developed in the context of a our work in Chapter 7, where all the participants authorized us to use their Facebook data (i.e., the content and privacy settings of the posts as well as the basic user profile) for research purposes. Further, to comply with Facebook's Platform Policies[3], we took extra care regarding the participants' privacy. Most importantly, the data will not be disclosed to third parties and the data collected represent the minimal amount of information needed in order to perform the experiments.

Dataset 1. The first dataset contains 45 users from 10 different countries. The users are all researchers and/or students in the field of computer science from the first authors' institution. We expect the data from these users to be trustworthy; the users are presumably more knowledgeable in using such digital platforms. From these 45 users, we collected all their posts, summing up to 26,528 posts (posts per user varies from 13 up to 3,176 posts). This dataset has been collected during February and March 2014.

Dataset 2. The second dataset has been collected using the CrowdFlower crowdsourcing platform, where the workers were asked to use the same Facebook app mentioned above. In this case, the authors are not personally familiar with the participants and therefore we have no a priori information on their knowledge of using social networking sites. Especially the former issue raised the concern of reliability, as there could be some workers who use fake profiles to finish the task and get paid. As a remedy, we considered only the data from workers who have a Facebook account that exists for at least 4 years and who have posted at least 25 posts each year. Using the CrowdFlower platform, we ended up with a much larger dataset, including 649 users and 769,205 posts in total. The crowdsourcing task has been running only for one day on November 27, 2014. In Table 5.1, we summarize the characteristics of both datasets.

[2] http://www.l3s.de/~kawase/forgetit/evaluation2015/

[3] https://developers.facebook.com/policy/

Table 5.1. Datasets.

	Dataset 1	**Dataset 2**
No. of users	45	649
No. of posts	26,528	769,205
Avg. no. of posts per user	602.431	1,185.215
Variance no. of posts per user	545,343	5,484,176
Min no. of posts per user	13	100
Max no. of posts per user	3,176	30,715

5.4 Analysis

5.4.1 Data Analysis

The privacy settings in Facebook regarding the audience of a post can be one of the following five main alternatives:

- **EVERYONE**: This setting means that the post is public. Even non-Facebook users are able to see these posts.
- **SELF**: Only the user who created the post can see it.
- **ALL_FRIENDS**: Posts with this setting are visible to users who are friends with the post creator, and to the friends of those tagged in the post.
- **FRIENDS_OF_FRIENDS**: In addition to the friends, posts with this setting are also visible to friends of the poster's friends, and to friends of friends of those tagged in the post.
- **CUSTOM**: In this setting the user deliberately specifies a customized privacy setting that includes or excludes specific users or groups from the audience. This option is usually accompanied by the fields *privacy_allow* and/or *privacy_deny*. These fields list users or group ids. CUSTOM includes three sub-values:
    - **ALL_FRIENDS**: Posts with this setting are visible to all friends of the post creator, except to some users or groups that are manually chosen by the creator.
    - **FRIENDS_OF_FRIENDS**: Posts with this setting are visible to all friends of friends of the post creator, except to some users or groups that are manually chosen by the creator.
    - **SOME_FRIENDS**: Posts with this setting are only to specific users or groups that were manually chosen by the post creator.

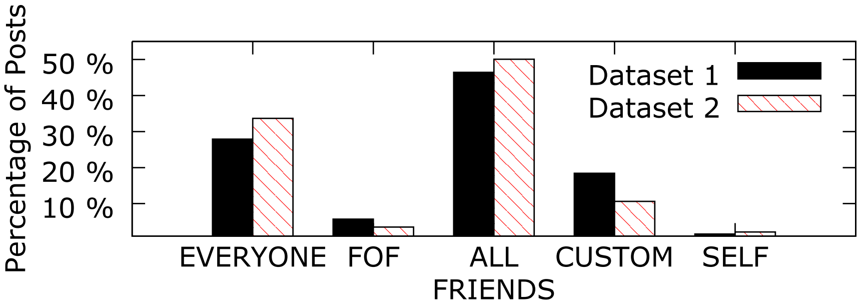


Figure 5.1. Distribution of the privacy settings for Datasets 1 and 2.

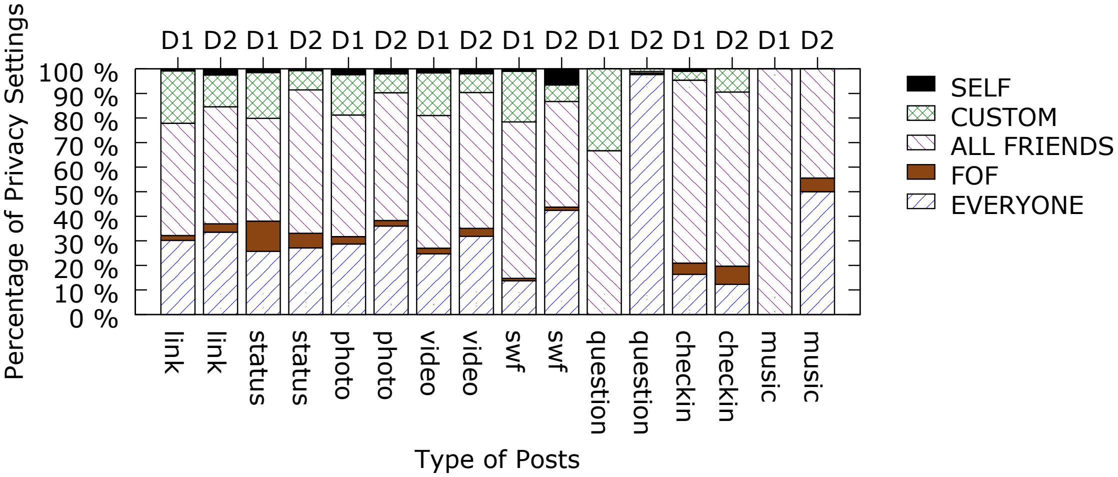


Figure 5.2. Distribution of posts normalized by post type for Dataset 1 (D1) and Dataset 2 (D1).

In Figure 5.1, we present the distribution of the alternative privacy settings for the posts in Datasets 1 and 2. We can observe some interesting patterns regarding the usage of Facebook privacy settings. First of all, for both datasets, we see a clear dominance of posts that are visible to all of the users' friends (45 to 50%) and of public posts (around 30%). We also observe a rather high demand for the option that denies access to specific users or groups (around 10% for Dataset 1 and 20% for Datasets 2), which is in line with our expectations from Section 5.1. This shows that quite often, users carefully 'hide' posts from particular users in their social networks.

Next, we provide a more detailed analysis taking into account the types of the posts. In total, there are eight different types that we identified in our datasets. Each type has unique characteristics that may influence the users in choosing the appropriate privacy setting. In Figure 5.2, we plot the distribution of privacy setting over post types. We see that especially post types of *music* and *Flash content* (shown as *swf*) are the ones that are shared with more general audiences (i.e., EVERYONE and FRIENDS_OF_FRIENDS), whereas post types like *link*, *status*, *video* and *photo* are more likely to be visible to restricted audiences (e.g., with privacy setting CUSTOM). As another interesting observation, almost all the posts of type the *question* (97.79%) are public in Dataset 2 (note that, while the situation is different for Dataset 1, we notice that there are only two posts of type *question* in this dataset, and hence findings are not representative). This is because of the fact that the privacy settings for *Questions* are not directly chosen by the user. In Facebook, *Questions* can only be posted in *Groups* or in *Events*, and the privacy settings are inherited from them. If a question is posted in a public group or in a public event, question is considered public (EVERYONE), and the creator of the post is not given the option to change it.

We also make an analysis of our datasets from the perspective of users. While doing so, we report the findings for Dataset 2, as the number of users is considerably smaller in Dataset 1 (though similar trends are observed for Dataset 1 as well). In Figure 5.3, we report the percentage of users who have posts with certain combinations of privacy settings. For instance, almost 43% of the users have posts from four different privacy settings, EVERYONE, ALL_FRIENDS, CUSTOM and SELF. Similarly, another 34% of users have posts from all the privacy settings. In general, the distribution in Figure 5.3 implies that these users are not unaware of the privacy setting, and indeed they intentionally use different privacy settings for their different posts.

For a deeper insight, we investigate how often users change their privacy settings in different posts. We performed a temporal analysis on the posts of each user to compute how many times she changed her settings; i.e., the number of times a user selects a different privacy setting for her post than that of the preceding post in chronological order. Figure 5.4 depicts the percentages of such changes for each user as shown on the x-axis. On the average, the users choose different privacy settings in 10.8% of the posts. This further supports our previous finding showing that at least some users deliberately choose different privacy settings. Given the fact that choosing the privacy settings is a task that is frequently triggered, and that the decision is quite often varying, we believe that users could benefit from tools that suggest the appropriate settings. Therefore, in the next section, we present a first step in the direction of predicting privacy settings.

5.5 Privacy Prediction Experiments

The data analysis in the previous section suggests that there might be dependencies between the privacy settings of a post and some characteristics of the the post or the user who wrote the post. In this section, we investigate whether it is possible to automatically predict a privacy setting for a post. Such a predictor can be used for recommending the most appropriate privacy setting to the user at the time of posting, and hence help to avoid cases of information leaking as exemplified before.

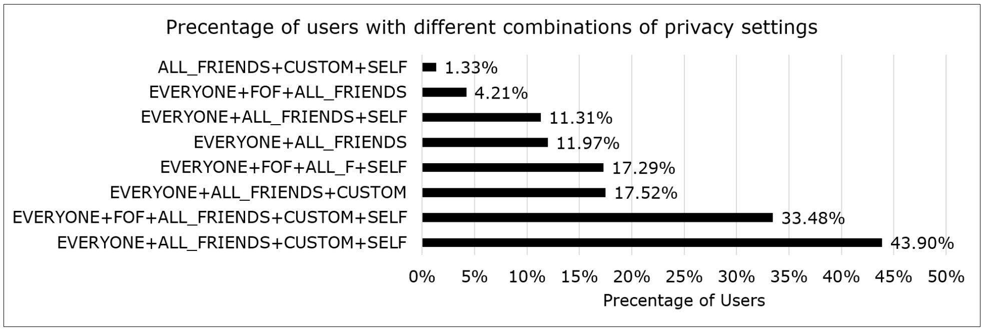


Figure 5.3. Distribution of users by their privacy settings combination (for Dataset 2).

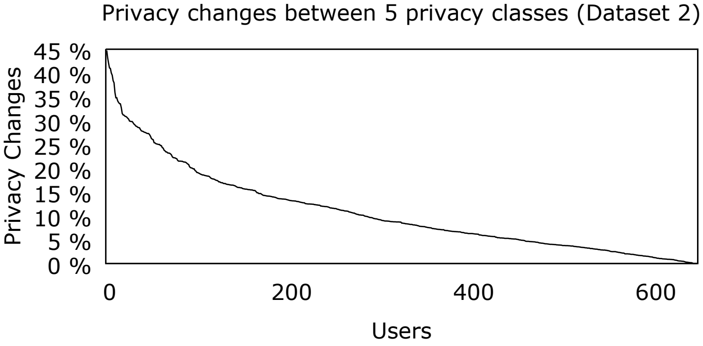


Figure 5.4. Distribution of the privacy changes by each user (for Dataset 2).

5.5.1 Experimental Setup

Target classes. To build a predictor with reasonable accuracy that can be employed in a practical setting, we opt for building a binary classifier and predicting whether a post has low or high privacy at an abstract level, rather than assigning each post to one of the privacy levels described in Section 3. We assume that posts that have the privacy setting EVERYONE or FRIENDS_OF_FRIENDS are in the class *Low_Privacy*, as they are visible to a very general audience. In contrast, the posts with the setting ALL_FRIENDS, SELF and CUSTOM are said to be in the class *High_Privacy*, as the user has the intention of sharing the post with a specific audience, i.e.; with only her friends, which can be the most typical case in a social platform, or even with a certain subset of them.

Dataset. For our classification experiments, we employ the crowd-sourced dataset (Dataset 2) that includes a reasonably large number of users and postings and, hence, can yield generalizable results. From the latter dataset, we discard all non-English

Table 5.2. The list of features used for the privacy prediction task.

Feature	Description	Feature	Description
Post metadata		**Context**	
has(message)	post has a message	sendFromMobile	post sent from an mobile application
length(message)	length of the message	dayTimes	(morning, afternoon, evening, night)
norm(length(message))	length normalized per user	sendAtWeekend	post sent during weekend
has(story)	has a story	**Sentiment**	
length(story)	length of the story	negative	the negativity score of a post
norm(length(story))	length normalized per user	positive	the positivity score of a post
has(description)	has a description	objective	the objectivity score of a post
length(description)	length of the description	**Users**	
norm(length(description))	length normalized per user	no_posts	total number of posts of a user
has(link)	post includes a link	no_friends	total number of friends of a user
has(icon)	post has an icon	gender	gender of the user
has(caption)	post has an caption	age	age of the user
type	type of post	country	country of the user
status_type	status_type of a post	education	the education level of the user
icons	describes user activity	**Keywords**	
tagged users	users tagged in a post	words_family	contains word from the list
Word vector		words_friends	contains word from the list
bag of words	top-1000 words using tf/Idf	words_work	contains word from the list
		words_holiday	contains word from the list
		words_travel	contains word from the list

posts using the language detector tool provided by [Shu10], as we aim to construct features based on the post content. Furthermore, for each user, we label the posts as Low and High Privacy; and get all the posts in the class with smaller number of instances, and undersample the posts from the other class. This is to obtain a balanced dataset (as the dataset is otherwise skewed in various ways; some users have a large number of posts, and furthermore, they are biased for a certain privacy class only). At the end, our dataset includes a total of 93,460 posts from 469 users; with an average of approximately 100 posts from each class, per user.

Features. In our experiments, we use features from six different categories (see Table 5.2). First, we have *metadata* features obtained from a post, such as the type of the post (e.g., link, photo, status, video, etc.), whether the post includes one or more of the predefined Facebook fields (such as message, story or description) and its length, and number of tagged users in the post. The *context* features capture the platform and time related information. From the post content, we first extract *sentiment* features, i.e., the positivity, negativity and objectivity scores computed using a vocabulary based sentiment analysis tool, namely, SentiWordNet [ES06]. The *keyword* feature category captures whether a post includes a keyword that might be related to a certain concept like family, friends, work, travel, etc. Note that, for each of the latter concepts, we manually compiled a small list (up to 20 words) of representative words. Another feature category is the *word vector*, i.e., the entire content of the post as a bag of words, as typical in text classification. We keep top-1000 most words with the highest tf-idf scores in the word vector. Finally, we have the *user* features, such as the number of posts and friends, gender, age, country and

Table 5.3. Classification results using all the features.

Naive Bayes

TP Rate	FP Rate	Precision	Recall	F-Measure	AUC	Class
0.640	0.255	0.715	0.640	0.675	0.780	LOW_PRIVACY
0.745	0.360	0.674	0.745	0.708	0.780	HIGH_PRIVACY
0.692	0.308	0.694	0.692	0.691	0.780	Avg.

REPTree

TP Rate	FP Rate	Precision	Recall	F-Measure	AUC	Class
0.810	0.191	0.809	0.810	0.810	0.887	LOW_PRIVACY
0.809	0.190	0.810	0.809	0.809	0.887	HIGH_PRIVACY
0.809	0.191	0.809	0.809	0.809	0.887	Avg.

education (the latter is obtained from the crowdsourcing platform). All the features in these categories are concatenated to obtain a single instance vector, i.e., applying the early fusion approach for different types of features (e.g., see [SWS05]). Note that since our predictor is to be employed during the post creation time, it is not possible to use typical social network features based on community feedback (e.g no. of likes, no. of comments etc.) employed in other contexts [COA14].

Classifiers and evaluation metrics. We apply the well-known classification algorithms NaiveBayes[JL95] as well as a fast decision tree learner, REPTree [WF05][Bre96]. For both algorithms, we use the implementation provided by the WEKA library[4]. For the evaluation, we use well-known measures from the literature: the true positive rate (TPR), false positive rate (FPR), precision, recall, F-Measure, and area under the ROC curve (AUC). All the reported results are obtained via 5-fold cross-validation. Remarkably, this implies that the posts of a particular user are distributed to training and test sets at each fold; and hence, the model will learn to predict the privacy based on not only other users previous decisions, but the user's own decisions, as well.

Results and Discussions. In Table 5.3, we compare the prediction performance for NaiveBayes and RepTree classifiers. The average TPR (i.e., accuracy) of the NaiveBayes predictor is 0.692, which is better than the random baseline with 0.5 accuracy (as we have a balanced dataset). Moreover, when predicting the High Privacy class, the classifier has a higher TPR (i.e., 0.745). This is useful in practice, as predicting a highly private post as public is more dangerous (as these are the cases where the information is exposed to a larger audience than intended) than vice versa. The overall performance of the RepTree classifier is even more impressive, as it yields an accuracy of 0.809 for both classes (and, on the average). For this classifier, average F-measure and AUC metrics are also over 0.80. These findings reveal that it is pos-

[4] http://www.cs.waikato.ac.nz/ml/weka/

Table 5.4. Classification results for each category of features.

REPTree						
Feature category	TP Rate	FP Rate	Precision	Recall	F-Measure	AUC
Word vector	0.715	0.285	0.719	0.715	0.714	0.793
Post	0.641	0.358	0.642	0.641	0.640	0.709
Users	0.600	0.400	0.601	0.600	0.598	0.673
Sentiment	0.591	0.408	0.593	0.591	0.588	0.652
Context	0.583	0.417	0.588	0.583	0.577	0.634
Keywords	0.553	0.446	0.553	0.553	0.553	0.592

sible to predict the privacy class of a post with good accuracy, and such a predictor can serve in suggesting the privacy setting of a post when it is first created.

Note that, since our dataset includes different numbers of posts from each user (but with the same number of instances from each class), it is also interesting to investigate whether classification performance is biased for the users who have more posts than the average. To this end, we filtered the dataset used in previous experiments, so that each user remaining in the dataset now has exactly 100 posts (50 from each class). In this new setup, the accuracy of the RepTree classifier is still 0.788, which implies that the accuracy can improve with more training instances from a particular user. Nevertheless, even for the case of 100 posts per user, the prediction accuracy is high (note that the scores for the other evaluation metrics are also similar and not reported here for brevity).

Finally, for the RepTree classifier reported in Table 5.3, we further investigate the performance of each feature category in isolation. Table 5.2 reveals that keyword features and word vectors are the least and most useful features, respectively. It is further remarkable that the classifier that use all features in combination perform considerably better than those based on a single feature category.

5.6 Conclusion

In this chapter, we presented an approach for supporting users in selecting adequate privacy settings for their posts. This work is based on a thorough analysis on privacy settings on social networks, particularly in Facebook. Our analysis shows that users customize their privacy settings quite often: for roughly one out of ten posts, a new privacy setting is chosen over time. The data also has shown that the type of post has a significant impact on the the choice of privacy settings. While posts of the type 'music' and 'question' tend to have a larger (less restricted) audience, 'status', 'photo' and 'video' are more often restricted to a smaller audience.

Targeting a supporting tool that could suggest users preferable privacy settings, we performed experiments for the privacy settings prediction task. By relying on different categories of features that can already be identified at the time of post composition, we were able to achieve a very good prediction performance with a recall and precision of more than 80% on average.

Additionally, our analysis demonstrated clear differences in users' behavior with respect to privacy settings. We observed that there are some users who are very sloppy regarding privacy settings, having most of their posts publicly available and not changing the settings. We also observed users who very often customized their settings, and users who prefer sharing data mostly with their friends. This difference in behavior indicates that a personalized model for privacy prediction might improve the already good results of the experiments presented in this work. A possible future direction would be to to collect more contributors (users) willing to collaborate with this research, and to build personalized methods for privacy settings prediction.

6

Recognizing Skill Networks and Their Specific Communication and Connection Practices

Social networks are a popular medium for building and maintaining a professional network. Many studies exist on general communication and connection practices within these networks. However, studies on expertise search suggest the existence of subgroups centered around a particular profession. In this Chapter, we analyze commonalities and differences between these groups, based on a large set of user profiles. The results confirm that such subgroups can be recognized. Further, the average number of connections differs between groups, as a result of differences in intention for using social media. Similarly, within the groups, specific topics and resources are discussed and shared, and there are interesting differences in the tone and wording the group members use. These insights are relevant for interpreting results from social media analysis and can be used for identifying group-specific resources and communication practices that new members may want to know about.

6.1 Introduction

People who work in similar professions typically share particular skills. Further, if people are asked to indicate their skills, it is expected that the skills they mention vary in granularity. For example, someone working in public relations may indicate skills in social networking and marketing, but also specific skills such as DTP software, writing press releases and time management.

It is also known that people from different professions or cultural backgrounds have different practices in how they communicate with one another, the communication mechanisms that they choose and the topics that they discuss [HCC11]. These differences can also be observed on a more private, personal level: programmers are usually more informal than bankers, people working in public relations are typically more active in social media than investors, and pastors will most likely talk about different topics than real-estate agents.

In this chapter, we investigate differences in communities within self-reported skill networks. We are particularly interested in discovering differences in their communication practices: how well is a professional community connected, how often do people post updates via Twitter or Facebook, what are the topics that they talk about, and what is the overall tone or sentiment of these communications? Particularly for people who aim to identify and approach experts from a different profession, who wish to promote their services in other communities, or who consider a career switch, it is important to know the unwritten rules in a network. For example, what would programmers think of overly positive marketing language? How often can one repeat an announcement? Would it be a good idea to add a personal touch or will that be considered "unprofessional" ?

Being aware of differences between professional communities is also important for interpreting statistical data from social network analysis. For instance, in some communities the average number of followers is considerably higher than in other communities. As a consequence, a person from a well-connected community like online marketing with, say, 300 followers, may be considered isolated; for a programmer, this is actually a very good number. The same differences apply for interpreting centrality and other in- and out-degree measures.

The main contributions of this study are: we provide an overview on how skills in professional networks are related and categorize these skills into professions. Further, we show to what extent different professions differ from one another in terms of connections, topics, sentiment and shared content. Finally, we discuss implications for social network analysis and the design of professional networking sites.

The remainder of this Chapter is structured as follows. In the next section we discuss related work, followed by a description of the dataset we used. In Section 6.4 we discuss the structure of the skill network derived from LinkedIn profiles and how this structure is reflected in the professions that we extracted using LDA. The results are presented in four subsections, covering: connections between people, topics that people discuss about, subjectivity and polarity of the wording, and the resources that they share. We end the chapter with a discussion and concluding remarks.

6.2 Related work

This work draws upon two related strands of research. First, our focus on skill and expertise networks fits in the research area of automated expert finding, in which both explicit and implicit information is used for identifying experts in a particular area. Our interest in differences between expertise domains in how people connect and communicate online follows the tradition of social media analysis.

Yimam-Seid and Kobsa [YSK03] argue that for the effective use of knowledge in organizations, it is essential to exploit tacit knowledge that is hidden in various forms, including in the people's heads. The authors also separate the need for 'information' from the need for 'expertise': the need for people who can provide advise, help or feedback - or who can perform a social or organizational role. Their expertise recommender made use of a hand-tailored expertise model.

MacDonald et al [MHO08] indicate that, in order to identify experts, documentary evidence is needed. This evidence may be based on documents, emails, web pages visited, or explicitly created profiles with an abstract or a list of their skills. This evidence will than be ranked with respect to a given query or goal skill profile. Based on the TREC W3C and CERC test collections, they evaluated to what extent additional evidence could improve expert retrieval. They found out that the proximity of a candidate name to query terms and clustering of main expertise areas are the best indicators. Extracted text from homepages and the number of inlinks did not have much influence.

Balog and De Rijke's experiments [BdR06] with data from the 2005 TREC Enterprise track show that user expertise can effectively be derived from email content; the persons being cc'd in an email were often authorities on the content of the message. Ghosh et al. [GSB+12] leveraged social media (Twitter) content for seeking experts on a topic. Their results indicate that *endorsement* in other users' Twitter Lists (of which the topics need to be extracted) infers a user's expertise more accurately than systems that rely on someone's biography or tweet content.

Guy et al. [GAC+13] examined indicators for expertise and interest as expressed by users of enterprise social media. The results are based on a large-scale user survey. They separate "expertise" (being knowledgeable or skilled) from "interest" (curiosity, basic knowledge, desire to learn more). As expected, interest and expertise ratings are correlated, with values for interest higher than for expertise. Results indicate that blogs and microblog provide different, more useful, information than communities and forums.

The above-mentioned studies suggest that people's skills and expertise can be derived from both explicitly provided lists and from their connections and communication patterns. This is consistent with Cingano et al. [CR12] observation that better-connected unemployed individuals, particularly those whose contacts were employed, are more easily reemployed. However, all of these studies were conducted in a single professional area or they generalized results between different areas. It is likely that considerable differences can be found between communities. For example, Hong et al.[HCC11] found that Twitter users of different languages adopted different conventions with respect to the inclusion of URLs, hashtags and mentions, as well as on replying and retweeting behavior. The main conclusion they drew is that the "average" behavior of the English-speaking community does not necessarily translate to other communities.

In our study, we will look at differences in how people from different professions are connected, the topics that they discuss, the subjectivity and polarity in their wording, and the type of resources or websites that they share. These topics have been subject of research in various studies, a small selection of them is discussed in the remainder of this section.

Kumar et al. [KNT10] analyzed the structure and evolution of online social networks. They showed that networks typically have one well-connected core region, but most users are located in one of several more or less isolated communities around it. These communities are typically centered around one central person, and it is unlikely that two isolated communities will merge at some point. In the next section, we will see that the structure of our skill-based network matches these observations.

Abel et al [AHK11] compared different approaches for extracting professional interests from social media profiles. Results indicate that dedicated tag-based profiles and self-created user profiles are most suitable for this task. Twitter profiles are more diverse but also more noisy; this effect can be reduced by extracting entities from running text. In a follow-up study, Abel et al. [AHH+13] analyzed the completeness of user profiles in different social media. The outcomes suggest that user profiles in networking services, such as LinkedIn, are more complete than those in services like Twitter. Further, the topics that users talk about differs between channels, but the overlap in topics is higher between services that are used for similar purposes. It was also shown that combining information from different services was beneficial for tag and resource recommendations.

Siersdorfer et al. [SCNSP10] investigated the usefulness of comments, as perceived by YouTube users. They found out that positive comments were considered more useful than negative comments. Differences between categories were also found: for example, science videos receive predominantly objective comments, politics relatively many negatively rated comments, and music videos mainly attract positively rated comments. These findings suggest that different communities have different norms with respect to commenting - we expect that the same effect can be observed if one compares different professions.

6.3 Dataset

In order to create a sufficient dataset, we first collected a set of 94,115 public user profiles from About.me, using the crawling strategy employed by Liu et al. in [LZS+13]. About.me is a personal profile site where users can include all their social-web accounts. From each profile, we collected the users' LinkedIn, Twitter and Facebook accounts.

For LinkedIn, the crawler gathered the public profile data, including skills and expertise tags, industry, job and number of connections.

For each account from Twitter, we gathered the complete user profile with information like number of followers and friends or number of lists the user is in. Beside that, we crawled the latest 200 Tweets using the Twitter Rest API [1]. The average number of Tweets posted by the users is 5,833, with a median of 1812. This indicates that most of our users are quite active in Twitter. We also had 33 users with more than 100.000 followers, which is already pretty influential.

The Facebook subset was collected using the Facebook API [2], which provides access to the public profile information of the users. Here, our crawl was focused on the Facebook timeline of the user, which mainly contains the shared posts. On average, the number of posts per user is 210 (median 23), with 34 users having more than 5,000 posts. Further, we collected data on the most popular features in Facebook, including the number of likes, number of comments, and number of shares on the users' posts.

In total, we have 33,516 users with a LinkedIn profile, 46,799 users with a Twitter profile and 34,523 users with a Facebook profile. Since the LinkedIn account serves as a source for our topics describing the users, we use for our analysis only Twitter and Facebook profiles that have a corresponding LinkedIn profile, resulting in a final set of 7,740 users. Our datasets are inherently noisy, as they represent human behavior. For example, the skills from LinkedIn are self-reported. Similarly, tweet content and Facebook posts are a mix of - among others - work-related announcements, private updates, and responses to others. However, this noise is reduced by the fact that our analysis is based on a fairly large collection of users.

6.4 Skill networks

LinkedIn users can list their skills in their profiles. It is a reasonable assumption that basic, more generic skills - such as 'management' - are more often mentioned than more specific skills - such as 'competitive analysis'. Further, one would expect that related skills - such as 'search engine optimization' and 'Web analytics' are often mentioned together, and that subskills are connected to one or two more generic skills - for example, 'Microsoft Word' would be often mentioned together with 'Microsoft Office' and 'Creative Writing'.

To verify whether these assumptions hold in LinkedIn, we visualized the network of skills using the graph visualization software Gephi [BHJ09] - see Figure 6.1, using a force-based layout, with the edge weights determined by how often skills are mentioned together. The four inlays that show parts of the network confirm the above-mentioned assumptions.

The largest node in the network is 'Social Media', which suggests that our sample is dominated by people who are professionally active in social media. Further, the areas surrounding the 'social media hub' have clearly defined sub topics. Top-right from social media are skills that are related to blogging and writing - with a subgroup

[1] https://dev.twitter.com/docs/api

[2] https://developers.facebook.com/docs/graph-api/

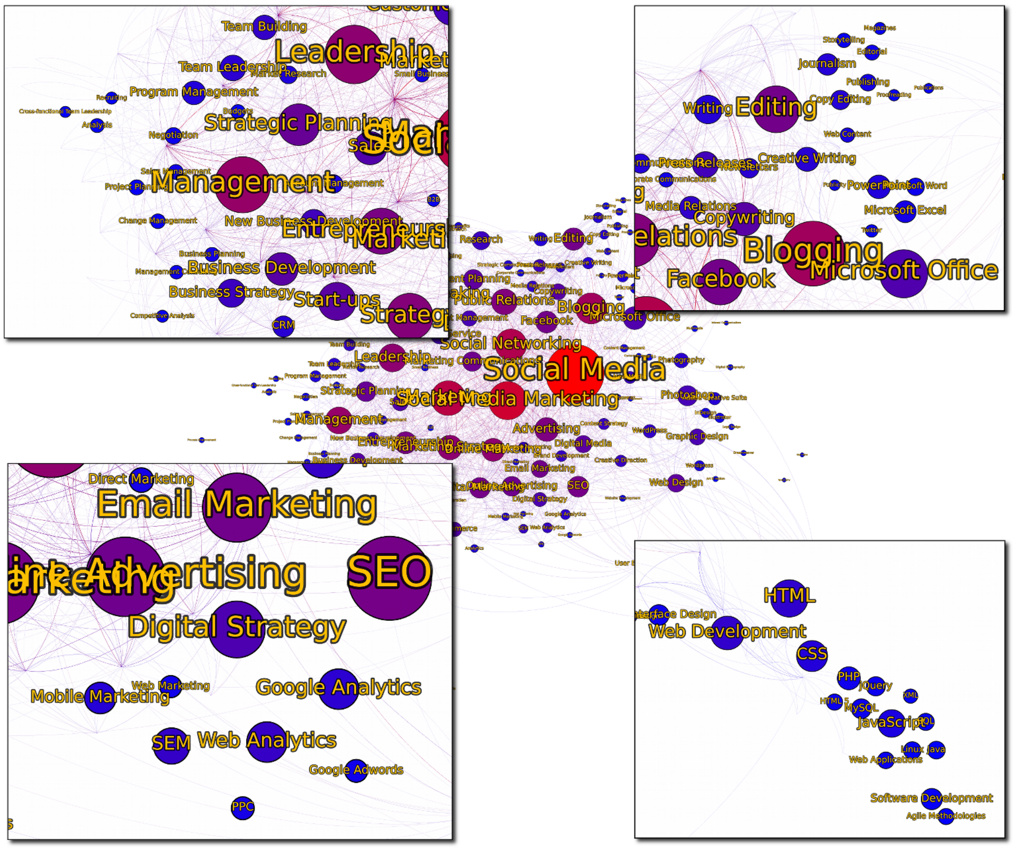


Figure 6.1. Skill network in LinkedIn. Larger nodes are more often mentioned. Skills that are often mentioned together are closer to one another. The four inlays are close-ups of parts of the network.

of graphic design skills. The more technical professions, such as web design and programming are located bottom-right. ‘Search engine optimization’ forms the bridge to the more marketing-related skills in the left part of the visualization. Top-left is dominated by more traditional management skills, including team building and planning.

6.4.1 Subgroups in skill networks

Table 6.1. The manually assigned topic labels and the most probable top-10 terms (assigned by the LDA method) for the 50 “Skills and Expertise” (SE) topics.

Topic Label	Top-10 Topic Terms
E-commerce-Strategy	marketing media social digital online strategy advertising analytics web management

Table 6.1. The manually assigned topic labels and the most probable top-10 terms (assigned by the LDA method) for the 50 “Skills and Expertise” (SE) topics.

Topic Label	Top-10 Topic Terms
Marketing-Strategy	research analysis strategy market product business development strategic competitive innovation
Social Media-Public Relations	media social creative public relations writing editing blogging press releases
Graphic Designer-Hands-On	design creative graphic direction art adobe suite illustration graphics identity
System/Network Administrator	windows server security network administration microsoft system vmware linux networking
Entrepreneur-Startup	business development strategy management start ups strategic entrepreneurship marketing planning
Search Engine Optimization-Tech.	marketing google web analytics online seo search advertising optimization sem
Web Designer-Graphical	web html design css wordpress photoshop adobe development graphic suite
Technical Support-Helpdesk	os mac office microsoft windows computer support technical hardware networking
Game Designer	design game games animation interior architecture video computer development planning
Social Media-’Spammer’/Analyst	social google media facebook twitter wordpress marketing analytics microsoft blogging
Manager or consultant	development community management program writing public leadership outreach planning education
Data Analysis-Programmer	data analysis science engineering research statistics computer design modeling matlab
Customer Management-People	customer management service sales retail team training satisfaction problem solving
Public Relations-International	policy public international research political relations english analysis writing government
Marketing-Events,Press	communications media marketing relations social management public strategic corporate event
Sustainability-Focused,Green	environmental energy management sustainability engineering sustainable construction project awareness water
Software Engineer-Commercial	software management cloud computing enterprise architecture data business integration saas
Financial Analyst	financial management analysis insurance finance planning business banking accounting risk
Marketing-Branding	marketing strategy media digital creative advertising brand social development online
Team Manager,Management	management team planning business project leadership development negotiation analysis strategy
Pastor-Church	pastoral church ministry youth leadership theology preaching studies development teaching
Professional Microsoft Product	microsoft office excel word powerpoint customer research service photoshop management
Marketing-Generic	marketing management media strategy social development advertising online brand business
Medical (Psychiatrists and co)	health healthcare medical clinical research psychology medicine counseling management mental
Beauty Industry	de fashion en styling trend beauty dise merchandising care comunicaci
Marketing-Networking	social marketing media public management event planning relations speaking networking
Marketing-Creator/Blogger	content media social marketing management web digital strategy online development
Web Programmer (#1)	sql net server asp development web microsoft software visual javascript
Manager-Project Planner	management business project process analysis improvement strategy leadership team planning
Real Estate	real estate homes home buyers sales property residential properties investment
Education-Teaching	learning education teaching technology development design curriculum educational training instructional
Creative Writer-Self-Employed	writing editing creative content publishing fiction copy blogging books articles
Web Programmer (#2)	development web html javascript css ruby java mysql php software
Journalist	journalism editing media writing news radio social style broadcast ap
Mobile Devices/Smart Phone	mobile product development devices applications strategy start web ups user
Film and Video Production	video production film editing final pro cut media television producing
Marketing-Generic Online	social media marketing networking online blogging digital web facebook design
IPR Person,Legal Analyst	law legal litigation property writing corporate intellectual research contract civil
Sales Manager	sales management business marketing development strategy selling product strategic account

Table 6.1. The manually assigned topic labels and the most probable top-10 terms (assigned by the LDA method) for the 50 "Skills and Expertise" (SE) topics.

Topic Label	Top-10 Topic Terms
Health and Lifestyle Advisor	coaching training sports wellness fitness nutrition health lifestyle weight personal
Music and Entertainment	music production audio sound theatre recording entertainment industry acting film
Photo Journalism-Art	photography art digital fine image painting portrait editing portraits photoshop
Hospitality and Tourism	food management hospitality event events travel wine tourism industry beverage
Software Engineer-Management	management software project testing agile analysis requirements quality assurance development
Training and Coaching	development management coaching leadership training business team organizational speaking change
Supply Chain Manager	security management military manufacturing supply chain operations engineering process improvement
Human Resources,Team Manager	skills problem solving communication team leadership thinking creative people building
Recruiter	recruiting management talent recruitment employee human search career resources sourcing
Usability Engineer	design user experience interface web information interaction usability mobile architecture

The skill network, as displayed in Figure 6.1, suggests that the LinkedIn network can be divided into skill-based groups, or 'professions'. As explained in the introduction, different professions are expected to have differences in terms of communication behavior, the way people are connected, the topics they talk about, the resources they use, and the way they express themselves.

In order to study topics beyond individual tags and to obtain more context-related information, we additionally employed Latent Dirichlet Allocation (LDA) [BNJ03] and modeled each LinkedIn Skills and Expertise tag-based representation of a user as a mixture of latent topics. For this, we used the LDA implementation in the Mallet library[3]. Given a set of term sets (users u_i represented by their Skills and Expertise tags in our case) and the desired number of latent topics, k, LDA outputs the probabilities $P(z_j|u_i)$ that the Skills and Expertise topic z_j is contained (related) in the user profile u_i. In addition, LDA computes term probabilities $P(t_j|z_i)$ for tags t_j; the terms with the highest probabilities for a latent topic z_i can be used to represent that topic. We empirically chose the number of latent topics as 50 for our LinkedIn dataset.

Table 6.1 shows the top-10 most probable terms for the 50 latent topics (called $professions$ in the next sections), as assigned by the LDA method. In addition, the table contains short topic labels which were manually assigned and will be used throughout the rest of this chapter.

[3] http://mallet.cs.umass.edu/

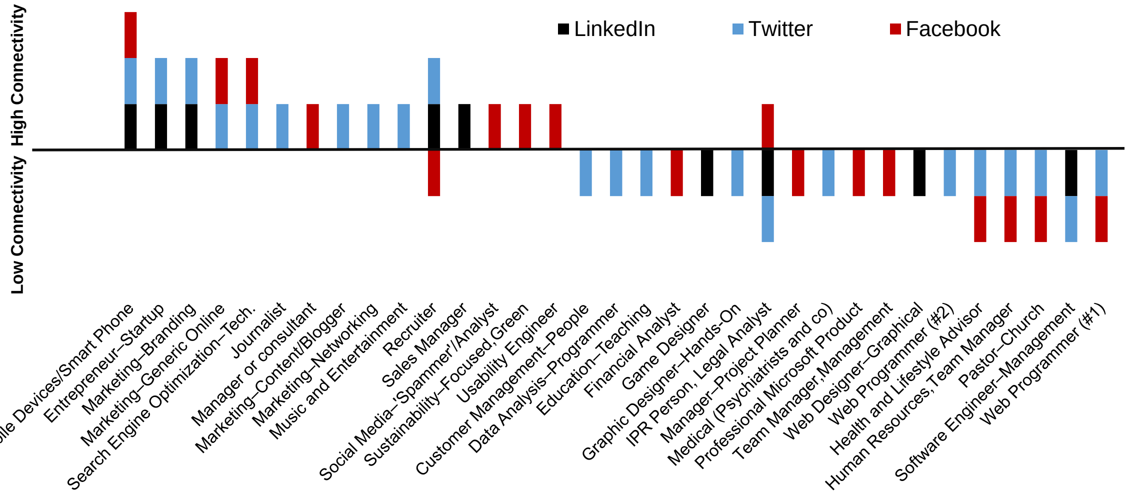


Figure 6.2. The professions with the highest and lowest connectivity for LinkedIn, Twitter, and Facebook.

6.5 Connections and activities

In this section, we discuss the results obtained from our analysis of differences in connections and activities between professions. We start with an overview of the differences in connections: which professions are better connected and more active. We continue with an analysis of the differences in topics that users post and tweet about: how generic or specific are these topics? Then we show that the differences in reasons why professions engage in social media have an impact on the sentiment and objectivity of the wording. Finally, we investigate which types of links and resources are shared in different professions.

6.5.1 Differences in connections

In this section, we look which professions are most and least connected with one another. Based on the insights obtained from the related work, we expect that professions that are in the core of the network are most connected and most active. In order to identify these differences, we took the following features into account:

- **LinkedIn:** We used the number of contacts as an indicator for the connections, no activity information was available.

- **Twitter:** Here, the user connections are based on the followers (incoming links), friends (outgoing links) and presence in lists (curated group of Twitter users). Activity is measured by the number of tweets.

- **Facebook:** As measure for connectivity, we used the number of likes, comments and shares (from friends) on the user's 'wall'. The number of posts of the users himself is an indicator of their activity.

For each of the social networks we created two lists of the top-5 highest and the top-5 lowest values on connections, presented in Figure 6.2. All professions displayed in this picture appear at least in one of these lists - all others are omitted.

As can be seen, several professions have high connectivity scores in more than one network. These include Mobile Devices (actually startups in this field), Entrepreneurs, Marketing and Search Engine Optimization. Low connectivity scores in more than one network are found among Web Programmers, Software Engineers, Pastors, Team Managers and Health and Lifestyle Advisors. In general, the left side mainly contains marketing-oriented professions, the right side IT-oriented professions and 'offline' professions.

In Twitter, the number of followers is highly correlated with the number of friends ($r = .79$) and presence in lists ($r = .87$). The number of followers depends less ($r = .59$) on the number of status updates. Within Facebook, no significant correlations between the number of (public) posts and likes or comments can be found - apparently, Facebook is less 'quantity-driven'.

Interestingly, apart from the marketing-oriented professions, the top-5 professions in terms of status updates (tweets) also includes Content Creators, Journalists and Pastors. These people probably use Twitter for announcements and 'spreading the word', even though - on average - they do not score very high in terms of followers.

6.5.2 Differences in topics

In order to compare what users of different areas talk about in different networks, we indexed the tweets and Facebook posts of the users into a Solr[4] Index. All the messages were processed trough a standard text processing pipeline, in which we removed stop words and used a stemming algorithm. Beside this, we also removed links from the text as we are only interested in the 'real words' used by a user. For tweets, we also removed the mentions of other users as well as the hash-symbol from hash tags.

This indexing allows us to compute the cosine similarity between different users and different professions. The similarity is calculated using the Solr 'more like this' functionality, which finds documents similar to a given document or a set of documents, based on the terms within the given document. These terms are selected based

[4] https://lucene.apache.org/solr/

on their TF/IDF values for the given document, which allows us to obtain a representative set of query terms for each user or profession. For all experiments, we selected the 500 most representative terms that occurred within at least five documents.

The different research questions we aim to answer with the experiments described in this section are the following:

- **Mentioning of skills.** Do people use Facebook and Twitter to talk about their professional skills as described in LinkedIn?
- **Similarity between networks.** How similar are users from the profession clusters in Twitter and in Facebook?
- **Specific & general topics.** Which profession clusters are very specific and which are very generic, based on the Facebook posts or Tweets?

To answer these questions, we built queries based on the terms used for the different skill and expertise groups (professions) from LinkedIn. Using these queries, we computed the score for the tweets or posts of every user. For each profession, we calculated the average score. The result of this computation is a matrix that shows how similar the users from the different professions are to the keywords of these professions. These matrices are shown in Figure 6.3. In order to make the differences better visible, we normalized the results for every query by dividing it by the maximum score. This ensures that the results are within a $[0, 1]$ interval and are comparable for every LinkedIn profession.

The diagonal lines in both diagrams show that most users use Facebook and Twitter to talk about their professional skills. In Twitter the diagonal is stronger than in Facebook, which indicates that Twitter is used for ‘professional’ communication to a larger extent than Facebook. Inside Twitter, we got an average self similarity (between the same profession cluster in LinkedIn and Twitter) of 0.884, while inside Facebook this values decreases to 0.741.

For answering the second question, how similar users behavior is in Facebook and Twitter, we indexed 50 users from each profession. We chose to use a similar amount of users per profession to remove the influence of the differences in cluster sizes. For each of the selected 2500 users, we computed the similarity to all other users based on the most representative terms used by the user in Facebook and in Twitter. The results are again two matrices, as shown in Figure 6.4. The matrix on the left uses the most common words in Twitter, the matrix on the right uses the most common words in Facebook. All values are normalized between 0 and 1.

Compared with the first two matrices, the first observation is that the diagonal is missing. This lack of within-cluster overlap indicates that users use Facebook and Twitter for different purposes. A remarkable difference between the two networks is the average similarity between random users: in Twitter, the average similarity is just 0.365 (the predominant green color in the left matrix); in Facebook, the average

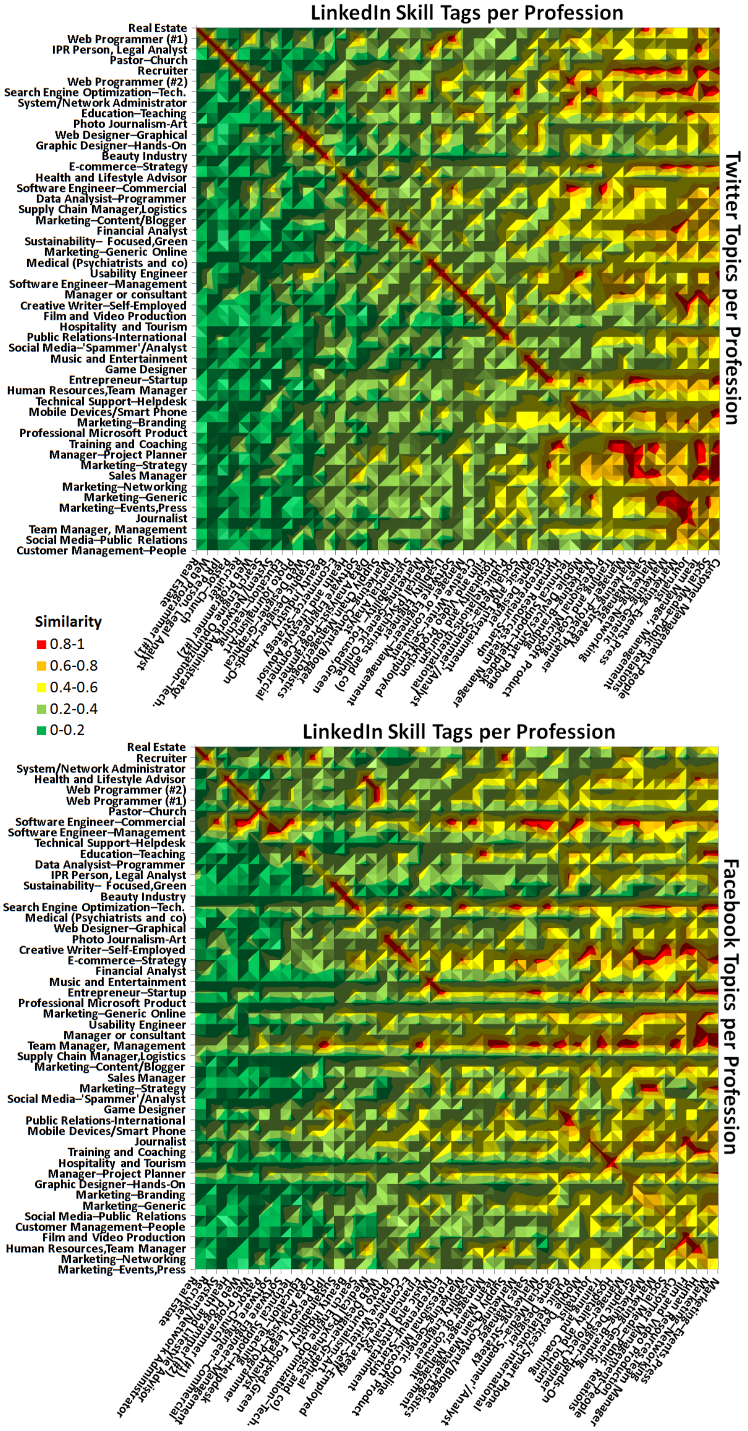


Figure 6.3. Similarity of skill tags from LinkedIn and terms used in Twitter (left) and Facebook (right). Similarities are summarized per profession.

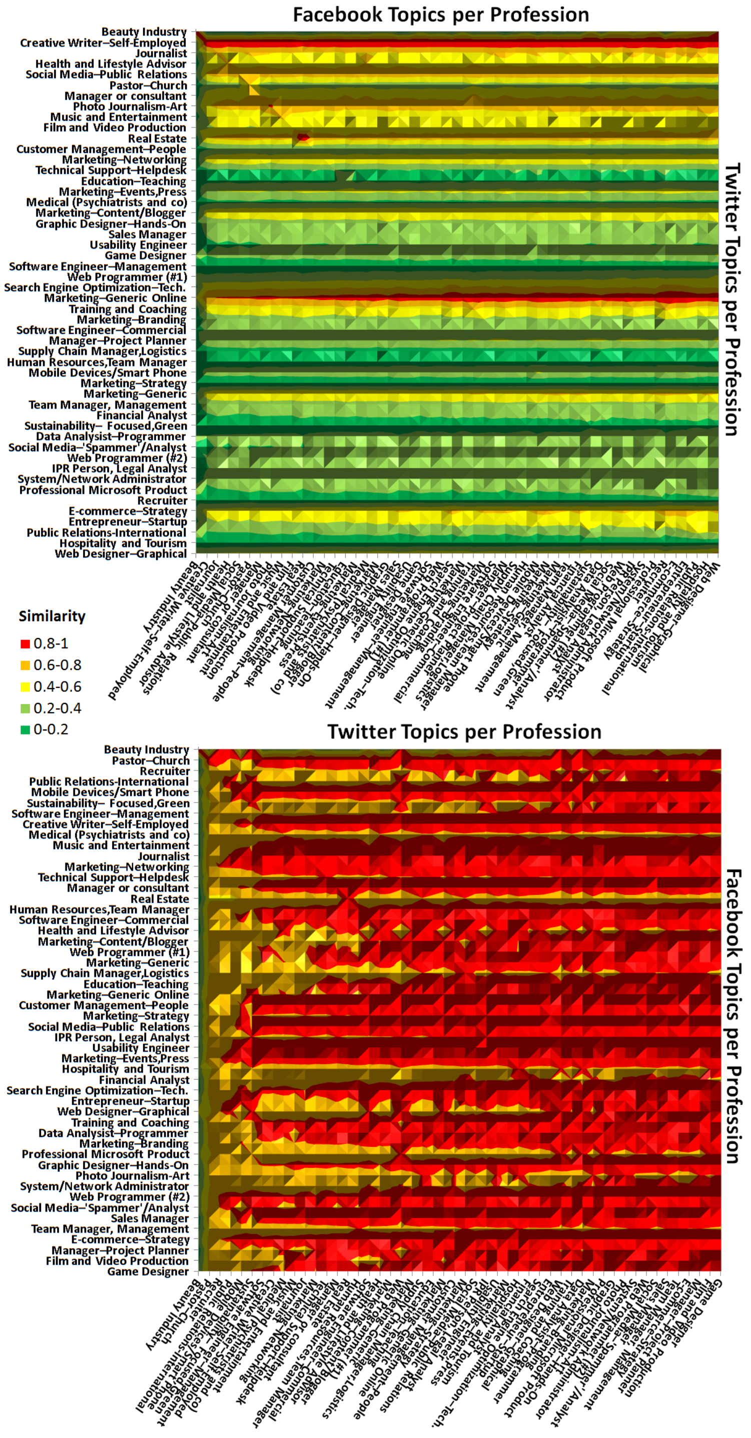


Figure 6.4. Similarity between topics that users talk about in Twitter (left) and Facebook (right), grouped by professions.

similarity is 0.818 (the predominant red color in the right matrix). The vertical lines in the left diagram indicate that some groups - particularly Creative Writing, Marketing and Social Media - write about very generic content within Twitter, while other groups use Twitter for more specific (professional) purposes. In summary, this indicates that Facebook is more general-purpose than Twitter, and that most profession clusters use Twitter for profession-specific purposes.

For analyzing the third question - which profession clusters discuss about more specific topics and which about more general topics - some first insights are already given by Figure 6.3, in which we ordered the LinkedIn profession clusters based on their average similarity to the users inside Twitter and Facebook. We see that professions related to Marketing and Social Media are listed on top in both diagrams, which indicates that the keywords used by these users are more generic and can be found in all professions. The bottom of both diagrams is dominated by technology-related professions as well as pastors, real estate and recruiters. Within these groups, the self-similarity is quite strong, which indicates that users within these professions exchange content-specific information.

We also indexed all messages from all networks and calculated the average similarity of one profession to all other professions, as shown in Figure 6.5. The blue bars show the similarity based on Facebook query terms and the red bars based on Twitter query terms. For some professions, like *E-commerce-Strategy* or *Usability Engineer*, we see large differences between the two networks. Other professions, like *Marketing*, *Journalist* or *Social Media*, are very general in both networks. The very general professions on the left seem all to be related to areas related to communication and marketing, the more specific professions on the right do not follow a clear scheme. Interesting to see is that many software-related topics are in the average area.

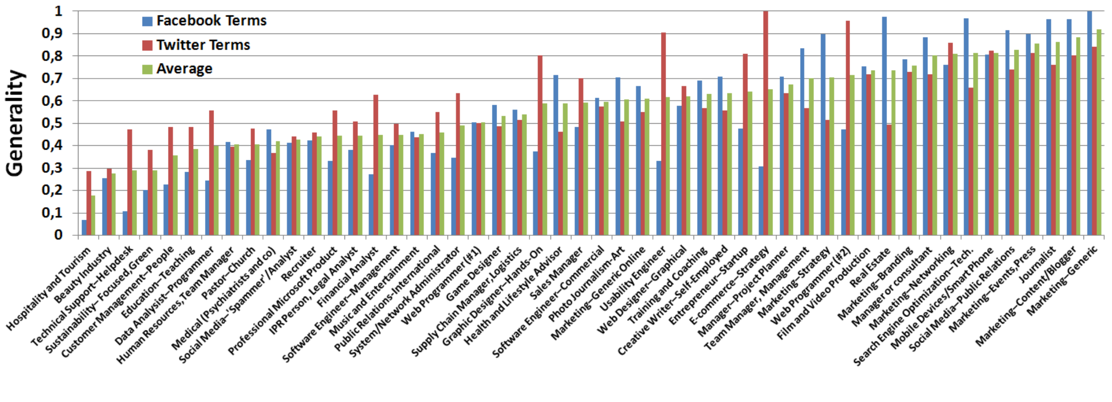


Figure 6.5. Comparison of generality of communication in different professions, based on terms from both Facebook and Twitter. Generality is the average similarity to all other professions.

6.5.3 Differences in sentiment

In this section, we use the SentiWordNet [ES06] lexicon to study the connection between the users' professions and the sentiment features of tweets and Facebook posts written by these users. SentiWordNet is a lexical resource built on top of WordNet. It contains triples of sentivalues (pos, neg, obj) corresponding to positive, negative or objective sentiment of a word. The sentivalues are in the range of [0, 1] and sum up to 1. For instance (pos, neg, obj) = (0.875, 0.0, 0.125) for the word 'good' and (0.25, 0.375, 0.375) for the word 'ill'.

We assigned a sentivalue to each tweet and Facebook post, in a similar manner as [SCNSP10, CASN13], where the authors analyse sentiment in short texts (YouTube comments and Web queries). Similar to the method used in these works, we restrict our analysis to adjectives, as we observed the highest accuracy in SentiWordNet. Finally, we computed the average positivity, negativity and objectivity over all tweets and Facebook posts that belong to a profession.

Table 6.2 shows the top-5 most positive, negative and objective professions with respect to user-expressed sentiments in Facebook posts and tweets. The users with skills in computer technical support and data analysis programmers tend to post the most negative messages in both Facebook and Twitter. Their posts or tweets often offer or request help for problems, i.e., *"@user Sounds like a hard drive issue. Either it's hitting bad sectors or the drive has literally slowed down and is having read/write issues"*. On the other side, users related to human resources, logistics and health, as well as lifestyle advisors post the most positive content in our collection. Some hand-picked examples from Twitter include *"Best food moments of 2013 #food http://t.co/JdYO36wVAY"*, *"Kids Eat and Stay Free at the Holiday Inn Washington DC. Bring the entire family for a holiday trip http://t.co/RhYa3zuHgu"*.

We also observed that users tend to be more objective in Twitter than Facebook, particularly for some of the professions. For instance, the average objectivity for *Pastors-Church* is up to 14% higher in Twitter than in Facebook. Many of the Facebook messages posted by users belonging to this profession express sympathy or commendation towards a religious topic or event, such as: *'2013 EVANGELICAL HEALING CONVENTION "Arise, Go, Preach" (Jonah 3:2)'* or a religious greeting *"May God bless your day as you display responsible actions and superior performance"*.

The differences in sentiment between the different skills and expertise groups may reflect that people in some professions are more positive or negative in general, or that they tend to formulate their messages more positively or negatively. Our interpretation, however, is that the differences in sentiment are largely caused by differences in intentions of tweeting.

The most positive groups are professions that use social media for selling and promoting items and events; it seems natural that these promotional messages are positive and motivating. On the other hand, the most negative group consists of

Table 6.2. Top-5 most positive/negative/objective professions w.r.t. user-expressed sentiments in Facebook and Twitter.

Facebook	**Twitter**
Positive	
Supply Chain Manager,Logistics	Human Resources,Team Manager
Technical Support-Helpdesk	Hospitality and Tourism
Medical (Psychiatrists and co)	Health and Lifestyle Advisor
IPR Person,Legal Analyst	Customer Management
Pastor-Church	Marketing-Branding
Negative	
Pastor-Church	Technical Support-Helpdesk
Technical Support-Helpdesk	System/Network Administrator
Training and Coaching	Social Media-'Spammer'/Analyst
Film and Video Production	Human Resources,Team Manager
Data Analysis-Programmer	Journalist
Objective	
Manager-Project Planner	Recruiter
Recruiter	Public Relations-International
Team Manager,Management	Team Manager,Management
Beauty Industry	IPR Person,Legal Analyst
Professional Microsoft Product	Education-Teaching

people who work individually on programming or writing tasks. We expect that these people mainly use social media for asking and providing help for problems and issues that they encounter. The least objective - or most subjective - topic groups mainly consist of people who provide advice and coaching in areas such as religion, health and lifestyle and entertainment. Most likely, these are people who aim to spread a particular message or opinion.

6.5.4 Differences in linked and shared content

Nowadays, a vast amount of content is shared by users through various social platforms. A recent study [Wil13] shows that 71% of online users have shared some type of content on social media sites. The most popular shared items usually refer to a picture, an opinion/status update or a link to an article. Another user study [Bre12] looks into the main motivation for sharing items, showing that most of the users (94%) carefully consider the usefulness of their shared content for the readers. While all of these recent studies imply the importance of users' shared content, there is no work that systematically investigates the link sharing patterns based on the users different expertise skills. We believe that our findings unleash the potential of analyzing

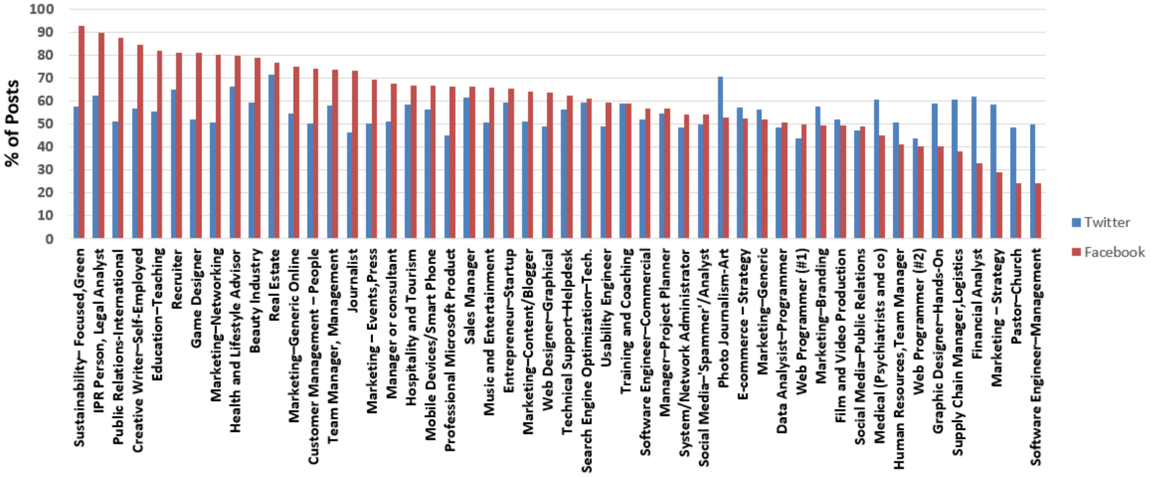


Figure 6.6. Percentage of Facebook posts and tweets sharing links, for each profession.

users' shared links, which is a rather overlooked source of information up to now.

In this section, we first provide an overview on the amount of link-based content shared by different experts in their tweets and posts. Next, we investigate the type of content shared by different experts, by looking into the main web-domains extracted from the shared links.

Figure 6.6 shows the percentage of Facebook posts and tweets that contain links for each profession group. In Facebook, 60.97% of the posts share a link. Users belonging to the *Sustainability-Focused,Green*, *IPR Person,Legal Analyst* and *Public Relations-International* professions are most likely to post links. In contrast, software engineers, pastors or market strategists are less likely to include URLs in their posts. In Twitter, Web programmers and software professionals attach links less frequently. On the other side, real estate experts, photo-journalists and health care advisors contribute with a considerably higher amount of links across their tweets. This is in line with our observation in Section 6.5.1 that these professions make use of social media for posting announcements. Overall, 54.76% from the tweets in our collection contain a link.

As an illustrative example, we computed ranked lists of web-domains from a set of tweets and posts belonging to the top-3 and bottom-3 most active web-domain sharers in our dataset. For ranking the resulting web-domain terms, we used the Mutual Information measure [MS99, YP97] from information theory, which can be interpreted as a measure of how much the joint distribution of features X_i (web-domain terms in our case) deviate from a hypothetical distribution in which features and categories (a specific profession versus all "other" professions, in our case) are independent from each other. Table 6.3 shows the top-10 web-domains extracted from the links shared within: 1) the posts written by users belonging to top-3 and bottom-3 Facebook profession groups, based on the link-sharing frequency and 2) the tweets written by users belonging to top-3 and bottom-3 Twitter profession groups, based on the link-sharing frequency.

Different profession groups tend to prefer linking different type of content across their messages. In our collection, the most shared links refer to a Social Network. For instance, in Twitter, Web programmers show a preference for *foursquare.com* (a platform for discovering friends' best locations), while real estate users link a vast amount of Facebook content. Note that, while Table 6.3 indicates noticeable differences in the preference towards different Social Networks, analyzing the underlying reasons for such differences is beyond the scope of this study.

At the same time, people tend to include links related to their expertise domains, i.e., *activerain.com*, *houselogic.com* for Real-Estate users and *arstechnica.com*, techcrunch.com for Web Programmers. For Facebook, we noticed that most of the shared web-domains seem to be less connected to the user's profession.

6.6 Discussion

In this chapter, we investigated differences in communication and connection practices between professions, as represented by the skill and expertise groups that we extracted from a representative dataset.

In our analysis, we used a combination of exploratory analysis, visualization and interpretation. These methods are not suitable for drawing strong conclusions on the exact structure and growth of communities and the interactions between the members. Among others, Kumar et al [KNT10] investigated these aspects as well. Our aim was to provide a complementary view on these structures and to give some insight in the people, professions and conversation topics that constitute these structures. Necessarily, these insights are partially given by means of representative examples. Keeping this limitation in mind, there are several key insights that can be drawn from the results.

In professional networks, connections between people based on shared skills follow the same structure as explicit connections, such as following, endorsing or befriending in social networks. The majority of mentioned skills are quite detailed and closely connected to a frequently mentioned more generic skill. By separating the skill network into clusters, skill and expertise groups - or professions - can be recognized.

The core of the skill network mainly consists of people who professionally use social media for specific purposes, such as marketing, promoting, branding and recruiting. These persons are typically well-connected, talk about common topics, share links from common resources and usually have a positive tone.

By contrast, several niche groups that are further away from the core are typically less connected and centered around a particular representative skill. Professions in

Table 6.3. Top-10 web-domains according to their Mutual Information values for tweets/posts written by users belonging to “One” profession vs. “Other” professions.

Top-10 distinctive web-domains for top-three professions, according to their % of links.

Twitter

Real Estate	Photo Journalism	Health and Lifestyle Advisor
facebook.com	facebook.com	facebook.com
foursquare.com	instagram.com	networkedblogs.com
youtube.com	etsy.com	youtube.com
paper.li	zazzle.com	graph.facebook.com
activerain.com	plus.google.com	articles.mercola.com
yelp.com	about.me	paper.li
trulia.com	vimeo.com	ebay.com
inman.com	blipfoto.com	amazon.com
houselogic.com	post.ly	about.me
scoop.it	fineartamerica.com	fitbit.com

Facebook

Sustainability Focused,Green	IPR Person, Legal Analyst	Public Relations International
facebook.com	facebook.com	apps.facebook.com
change.org	dangerousminds.net	nytimes.com
ulink.tv	politicususa.com	npr.org
elpais.com	addictinginfo.org	nyti.ms
youtube.com	huffingtonpost.com	youtube.com
avaaz.org	alternet.org	washingtonpost.com
librarything.com	thinkprogress.org	salon.com
europapress.es	forwardprogressives.com	behance.net
actuable.es	dailykos.com	change.org
zimbio.com	fab.com	i.imgur.com

Top-10 distinctive web-domains for bottom-three professions, according to their % of links.

Twitter

Web Programmer #1	Web Programmer #2	Profesional Microsoft Product
foursquare.com	foursquare.com	instagram.com
youtube.com	twitter.com	foursquare.com
fancy.com	youtube.com	twitter.com
getglue.com	i.imgur.com	twittascope.com
blogs.msdn.com	techcrunch.com	youtube.com
arstechnica.com	theverge.com	plurk.com
engadget.com	twitpic.com	justunfollow.com
path.com	twitter.yfrog.com	gofundme.com
fplus.me	plurk.com	runkeeper.com
techcrunch.com	meetup.com	infojobs.net

Facebook

Software Engineer Management	Pastor-Church	Marketing Strategy
youtube.com	apps.facebook.com	nike.com
apps.facebook.com	ludia.com	buff.ly
facebook.com	barackobama.com	youtube.com
nblo.gs	instagr.am	tripit.com
bbc.co.uk	gofundme.com	groupon.com
livingsocial.com	facebook.com	act.credoaction.com
ludia.com	eventbrite.ca	secure.sierraclub.org
meetup.com	amzn.to	generalassemb.ly
mashable.com	amzn.com	gr.pn
amazon.co.uk	itunes.apple.com	animoto.com

which (individual) productivity is more important than communication - such as programming and writing - seem to use social networks predominantly for specific purposes, such as providing or asking for help or feedback. Due to this different intention of use, the activity level, the topics discussed and the resources shared differ highly from what happens in “the core”.

These observations have clear implications for social network analysis, particularly for professional networks. Firstly, it is clear that averages for the whole population - and interpretation of these averages - are often only meaningful for the central core. The dynamics in subgroups are in many cases quite different - based on our qualitative evidence mainly caused due to differences in intention of use.

Zooming into the topics and links that are specific for a subgroup, and providing these to users who are new to the community or who aim to connect to it, seems to be a promising approach to get these users acquainted with the community and to get a feeling on the unwritten conventions and rules within these communities. In addition, the group-specific resources - such as technology-oriented websites - often serve as a useful starting point for exploring a new expertise area. These insights can be used as starting points for new browsing and search functionality in professional networking sites.

6.7 Conclusion

Within a skill network, several subgroups - or professions - centered around a particular skill can be recognized. Our analysis shows that these subgroups have specific unwritten conventions and rules, mainly caused by differences in intention for using social media. These insights call for separate analysis or treatment of activities within these subgroups, and provide several starting points for new functionality in professional networking sites.

7 Those Were the Days: Learning to Rank Social Media Posts for Reminiscence

Social media posts are a great source for life summaries aggregating activities, events, interactions and thoughts of the last months or years. They can be used for personal reminiscence as well as for keeping track with developments in the lives of not-so-close friends. One of the core challenges of automatically creating such summaries is to decide which posts to remember, i.e., consider for inclusion into a summary and which ones to forget. In this Chapter, we design and conduct user evaluation studies and construct a corpus that captures human expectations towards content retention. We analyze this corpus to identify a small set of seed features that are most likely to characterize memorable posts. Next, we compile a broader set of features that are leveraged to build general and personalized machine-learning models to rank posts for retention. By applying feature selection, we identify a compact yet effective subset of these features. The models trained with the presented feature sets outperform the baseline models exploiting an intuitive set of temporal and social features.

7.1 Introduction

Human memory is very effective in keeping us focused on relevant things by forgetting irrelevant information. However, we also quickly forget the details of events or do not completely and/or correctly remember them. This is especially true for episodic memory [Tul02], which is, roughly speaking, responsible for remembering the details of individual events. In episodic memory, the memories of new events interfere with older memories as an effect of proactive interference [Und57]. Furthermore, the memories of similar experiences blur into each other very easily, making it difficult to distinguish between the details of individual events (as an effect of retroactive interference [MM31]). Thus, the information collected over time in social media applications, such as, Facebook[1] can play an important role for complementing human memory: In the first place, it is created in near real-time and mainly for interaction, sharing and presenting oneself. However, if processed and presented in the right way, it can also be used to revive event memory and support reminiscence.

We are in an unprecedented situation where traces of everyday life and personal history is documented as a side effect of interacting with peers, no longer restricting life logging to major personal events or holidays. By documenting personal life, this information clearly constitutes an asset. Especially, the large volume of photos and videos created and shared by individuals today are considered a valuable part of personal remembrance [KS10]. In addition, recent work has shown the interest of users in using social media content for reminiscence and self-reflection as well as the potentials of social media content for this task. In [ZSN+13] for example, a study with Facebook users has discovered a considerable interest in managing a *personal region* for personal reminiscence and reflection about oneself. Facebook's own investment into its applications *Year in Review*[2], which aggregates selected content from the past year into a video, and *On this Day*[3], which presents a user her memories from that day in her Facebook history, highlights the importance and timeliness of the topic as well as the challenges involved[4].

In the light of the above discussions, we believe that harvesting a personal history from the vast amount of data in social media applications arise as an important and interesting research question. Such summaries are not only useful for personal remembering: they also provide an important source for catching up with what happened in the lives of not-so-close friends (e.g., former class mates), whose activities we do not have time or interests to follow on a day-to-day basis.

Automatically creating social media summaries, which meet human expectations on what to remember and what to forget is, however, a challenging task [KNS13, ZSN+13]. As the data involved in typical social media applications are in the form of posts (including text, video and/or audio) and interactions over such posts (such as likes, comments, shares, etc.), the key to create personal summaries automatically is deciding on the posts that need to be included in the summary, i.e., the *memorable* posts. Similar to the notion of relevance in information retrieval, it is not possible to exactly model the memorable as this is a highly subjective perception (and hence, a binary classification model is not likely to be useful); yet one can build models using a broad set of features to rank a user's posts (just like in document retrieval), with the goal of having the most memorable ones at the top positions. Such a ranked list would not only allow browsing of a user's past posts starting from the most memorable ones and scrolling infinitely, if the user has the time and will, but also creating a personal

[1] https://www.facebook.com

[2] https://www.facebook.com/yearinreview/

[3] https://www.facebook.com/onthisday/

[4] https://research.fb.com/facebook-memories-the-research-behind-the-products-that-connect-you-with-your-past/

summary from the top-ranked posts. Therefore, in this chapter, we introduce learning to rank for retention as a novel research problem and seek answer to the following questions:

- What are the features that may characterize the memorable posts?
- Can we build general and personalized models for ranking users' posts?
- Can we identify a subset of features that allow building compact ranking models that are as effective as the ones employing all the available features?

Our contributions in this chapter to address these questions are as follows:

- To investigate the first question, we designed user evaluation studies involving two complementing sets of participants: a small, yet known set of colleagues/friends (with 41 subjects) and a larger set of workers from a popular crowd-sourcing platform (with 470 subjects). In these studies, participants graded a subset of their own posts using a 5-point likert scale in terms of whether these posts are worth keeping for future needs, i.e., memorable, or not. Using this unique data collection, we first conduct a primary data analysis to investigate to what extent a small set of intuitively chosen features can characterize the memorable posts. We find that the post type and interactions on the post (i.e., number of likes and comments), together with the age of the content, seem to be the best ad hoc evidence to identify the post that may be worth to select for supporting reminiscence.
- While our manual data analysis allowed us to detect a small set of seed features, machine-learning based approaches for similar tasks (say, ranking models for search engines) typically employ all potential features (e.g., up to hundreds or even thousands [MSO12, YHT^{+}16]) that can be extracted from the data, as a feature that is found to be less useful on its own can improve the overall performance of the model when combined with other features. Therefore, we also compiled a broad set of 111 features from our data collection to capture the factors that might influence the multi-faceted retention decisions of users. By leveraging these features, we build general and personalized machine-learning models for ranking memorable posts. Since there does not exist a baseline set of features in the literature for the novel task of ranking for retention, we use the most promising features from our data analysis to train a competitive baseline and compare our models against the latter.

  Our experiments reveal that general models outperform the models with the baseline features and provide relative improvements of up to 16.8% and 20.3%, in terms of the nDCG@5 and nDCG@10 metrics, respectively. Furthermore, the range of the effectiveness scores for these models (i.e., an nDCG score of up to 0.64) is reasonable in comparison to state-of-the-art performance in typical learning-to-rank settings (optimized for relevance); e.g., nDCG@10 is reported to be 0.49 and 0.78 for the Microsoft and Yahoo learning to rank datasets, respectively, in [GLNP16], and it is less than 0.60 for ranking tweets in [DJQ+10]. This indicates that our approach in this chapter, i.e., training models to rank social media posts for retention, is appropriate and achievable.

To train personalized models, we used the k-nearest neighbors of a user (as in [GLQ+08]), and obtained moderate yet promising additional gains (i.e., up to another 2% relative improvement in nDCG@5) in certain cases.

- As our last contribution, we focus on feature selection in order to identify a compact yet effective set of the features that are most effective in ranking posts for retention. To this end, we apply a greedy feature selection method that is shown to perform well in learning-to-rank settings [GLQL07, CORA14]. We show that especially for the higher rank cut-offs, i.e., generating top-15 and -20 rankings of posts, the general models can be trained with a considerably smaller number of features (i.e., between 30 and 72 features instead of all 111) without any adverse effect on the effectiveness, i.e., nDCG scores, but even with occasional positive improvements.

The remainder of this chapter is structured as follows. In Section 7.2, we discuss the related work. In Section 7.3, we describe the user evaluation studies and present our data analysis. Section 7.4 describes the candidate features for retention and Section 7.5 presents the ranking experiments and our main findings. Finally, in Section 7.6, we present our conclusions and their implications for future work.

7.2 Related Work

Due to its popularity, Facebook is the focus of various research works. This includes studies on the usage of Facebook and on usage motivations [Joi08], analysis of changes of users behavior over the time triggered by the evolution of the Facebook application, such as the introduction of the the timeline [ZSN+13], as well as analysis work on the relations and social capital in Facebook [ESL11, EGV+13].

Studies about changes in the Facebook application and the way it is used are relevant for our work, since they have an impact on the material that is available for inclusion in the summaries. A study on Facebook usage, which is very important for our work, is the one presented in [ZSN+13]. Here the authors identify three regions of Facebook functionality, where the personal region is used for the management of personal data for themselves as a type of *personal locker*. The authors point out that due to the focus of Facebook on recent activities (e.g. timeline) the management of data from the past and the transition of data into personal region imposes several challenges. This is a clear motivation for our life summary approach, which can help the user to select and organize data for his reminiscence. In [SO13] the authors extract features based on the usage and network properties of the users for predicting the users motives for using Facebook. In our work, we adopt their features and extend them for our scenario. Since our work is, however, focused on posts we apply the metrics to the posts not to the user profiles (focus on social networks). To the best of our knowledge, there is no published research work on the summarization of Facebook posts. There is, however, the Facebook application *Year in Review*, which go in this direction.

Human Remembering and Forgetting. In [SLW11], a recent study has shown that a search technology, such as Google, effects on human memory. Similarly, shared retrieval-induced forgetting in a social network can reshape the memories of speakers and listeners involved in a conversation, so-called collective memories [CH12]. Typically, such studies shed a light on understanding how humans remember or forget information. This understanding can benefit methods that aim at complementing the human ability to remember or forget such as managed forgetting. When it comes to organizational and societal memory (and forgetting), we face difficult challenges to deal with - whether in the case of state archives detailing a dictatorial past, or sensational media reports that are subsequently shown to be false, and the unending digital memories they create [MS09]. In recent years, there have been several works addressing digital preservation from the Human-Computer Interaction (HCI) perspective, e.g., focus on the system design to support human memories [BP11, CNBM$^+$12, KW11]. An interdisciplinary model and approach for flexible and gradual managed forgetting in a digital memory has to be developed that meets human expectations and is driven by the goal of the digital memory complementing human memory. Supporting managed forgetting in a digital memory is a novel concept, for which no former experience and best practices exist. It is therefore important to thoroughly analyze the human expectation for this process. An interdisciplinary approach is planned for this purpose. The idea is to investigate, what we can learn from the way a human memory forgets and remembers. Humans are, for example, very effective in (a) rapidly extracting the general gist of an experience, while forgetting many details, in (b) extracting common features of similar experiences avoiding the “storage” of repeated features, and in (c) identifying data that are only temporally required and can be forgotten after task completion. Those and further characteristics of human forgetting will be further investigated. Selected characteristics will flow into a model for managed forgetting. The goal is, however, to complement not to copy or replace human memory. This perspective will create the highest benefit in the interaction of humans with digital memory. For analyzing the expectations towards managed forgetting user studies will be performed.

Personal Information Management (PIM). Our work is also related to the area of personal information management (PIM) [Jon08]. PIM tries to understand the best practice of users in storing, retrieving, and (re-)using information and to develop new methods and tools for this purpose. Originally, PIM mainly focused on information on a user’s desktop (and on non-digital information) and was subsequently extended to also incorporate activities in the Web (e.g., for search [DCC$^+$03]). Actually, the problem we are considering, on how to deal with the growing number of social web posts on the long run, brings social web applications closer to the typical personal information management problems [ZSN$^+$13]. Furthermore, the work on temporal organization of personal information in [KHF$^+$09] is relevant for our final goal of creating life summaries, since it investigates time driven organization and visualization of personal information such as *Personal Narratives*.

Personal Information Management (PIM) is about finding, keeping, organizing, and maintaining information [Jon08] both in a personal and organizational context. PIM is a vivid research area trying to understand the best practice of users in storing, retrieving, and (re-)using information and to develop new methods and tools to overcome their problems (e.g., personal information retrieval [DCC$^+$03], a temporal perspective in PIM [Jon10, KHF$^+$09]). Marshall identifies several issues in PIM with personal digital archives which start to pile lots of information over the years. users are deciding for deletion because it is a cognitive demand, assessing the value of information in advance because it is difficult to judge, and finally, “a full chronological and contextual record is essential for using one’s archives as a memory prosthesis” [Marshall, 2011]. A promising direction to support users in organizing personal information is the Semantic Desktop [SDE$^+$06] which introduces a knowledge representation layer to describe the information elements on the desktop (such as emails, webpages, documents, pictures) with a personalized vocabulary. This approach has been further extended to activity-based desktop search [CCNP06], and semantic search and ontology-based information extraction [GAS$^+$09].detection in the PIMO [Kubo et al., 2008], personal task management [Maus et al., 2011], personal image collections [Klinkigt et al., 2011], or bootstrapping from individual email [Schwarz et al., 2011].

In addition, our problem of identifying memorable posts can also be considered as a special information value assessment problem. Several valuation methods have been proposed, employing a rich variety of criteria. Many approaches take observed usage in the past as the main indication for information value, i.e., probability of future use [Che05].

7.3 User Evaluation Study

In order to better understand human expectations and build a ground truth of memorable social network posts, we set up two evaluation studies on top of the Facebook platform. The main goal of these evaluations was to collect participants’ opinions regarding retention preferences for their own Facebook posts from different time periods. The first evaluation is an extension of a preliminary study that has been described in [NKKN14]. For a deeper understanding of user expectations we conducted a second evaluation including a larger number of users recruited via crowdsourcing. In this section, we first describe these evaluation studies and then provide an analysis of the collected data.

7.3.1 Setup and Methodology

For encouraging and facilitating participation, we prepared an intuitive evaluation system in the form of a Facebook app. In order to participate, users have to log in with their Facebook credentials and grant the app the permissions to access some of their Facebook information, such a the profile, timeline, and friendship connections. After that, participants were presented with a running list of their posts.

During the evaluation, each participant had to judge their own posts on a 5-point Likert scale answering the following question: *Which posts do you think are relevant to and worth keeping for the future needs?* Once a post is evaluated (with a rating from 0 (irrelevant) to 4 (extremely relevant)), it fades out providing space for further posts to scroll up. The evaluation interface of a single post contains information about its author, creation date, description, image, etc.

Using the above framework, we conducted two evaluation studies that essentially differ in the number of participants and the way they are selected, as described in the following.

Evaluation Study-I

The first study was performed between the second week of November 2013 and the third week of February 2014. We had 41 participants, 24 males and 17 females, with age ranging from 23 to 39 years old. Participants were recruited through research communities, including colleagues from the authors' institutions, students, and their friends (and hence, we refer to collected data as the **Lab** dataset hereafter.) In this evaluation the participants were asked to judge about 100 to 200 of their posts. It is important to note that we are not judging participant's memory skills, but instead we are collecting their personal opinions regarding the retention preferences. Due to that, we presented participants' posts in a chronological order starting from the latest.

In total, the dataset includes 8,494 evaluated posts, essentially covering the period from 2014 back to 2009 (detailed statistics will be presented later). Additionally, once the users provided us authorization to access their data on the Facebook platform, we were able to collect general statistics that help us to depict their use of Facebook social network. We believe that this first evaluation study, despite a relatively small number of participants, is still interesting and worthwhile since it is ensured to be based on real users with real profiles, i.e., does not include untrustworthy participants, as can happen in the more uncontrolled setup described next.

Evaluation Study-II

In November 2014, we conducted a second evaluation with a larger number of participants from a popular crowdsourcing platform, CrowdFlower. The task for the workers and online evaluation system was the same as in the first evaluation. To begin the evaluation study, the workers had to follow a link to our system in the Human Intelligence Task (HIT) page at the crowdsourcing platform, and login with their Facebook account. Only those who had a Facebook account of at least 4 years old were allowed to participate (so that posts from a time span that is comparable to that of the first evaluation could be evaluated) and each worker had to evaluate at least 100 posts to complete the evaluation task. Each participant got 25 posts randomly selected from each year, from 2014 back to 2010. In cases where the users evaluated more than 100 posts or the Facebook profile of the user had less than 25 post for each year, they got older posts to evaluate. Overall, we ended up with the so-called **Crowd** dataset including 57,281 annotations from 470 users.

At the end of the evaluation task the participants were asked a few questions to collect personal information about their age, level of education, and country, in case that not all this information is available in their Facebook profile. After answering the questionnaire, the participant could complete the task by entering a code provided from our external evaluation website. On the average, the task was completed in 102 seconds. Note that, the pay per task was 5 cents, a reasonable amount for a simple task that does not require any background knowledge or skills and that took in average less than 2 minutes to complete. Further, it is worthy to mention that previous work [MW09] has demonstrated that higher monetary incentives does not necessarily improve quality in crowdsourced tasks.

As untrustworthy workers are not unlikely in crowdsourcing platforms (e.g., [GKDD15]), we applied additional measures to improve the quality of the collected data. In addition to enforcing the condition that each Facebook profile has to be at least 4 years old, we also cross-checked information provided in the questionnaire against that in the participant's Facebook profile to identify untrustworthy profiles, i.e., those with contradictory information. As before, the participants have allowed our application to access their profile information, timeline and their friendship graph on the Facebook platform. In total we ended up with 470 participants.

Dealing with Privacy Issues: In both user evaluation studies, we took extra care regarding the participants' privacy and to comply with Facebook's Platform Policies[5]. It is declared and guaranteed that collected data will not be disclosed to third parties. Furthermore, the data cached represent the minimal amount of required information for the experiments.

[5] https://developers.facebook.com/policy/

Table 7.1. Basic statistics for the Lab and Crowd datasets.

	Lab Dataset	Crowd Dataset
No. of users	41	470
No. of annotated posts	8,494	57,281
Avg no. of annotated posts (per user)	207.170	121.874
Min no. of annotated posts	12	100
Max no. of annotated posts	1,128	326
Female participants	17 (41%)	136 (29%)
Male participants	24 (59%)	334 (71%)
Age range of participants	23 - 39	18 - 65
Year of evaluation (duration)	2013 & 2014 (2 days)	2014 (5 days)

Table 7.2. Top-5 countries of the participants in the Crowd dataset.

Country	Percentage
IND	12.2
PHL	8.1
BGR	5.5
VEN	5.5
ITA	3.9
Others	64.8

Table 7.3. Educational level of the participants in the Crowd dataset.

Education Level	Percentage
Some high school (no diploma)	7.2
High school (diploma)	15.7
Some college (no degree)	18.2
BSc/MSc	44.1
Associate/Professional/Vocational/Tec. degree	14.3
Others	13.5

7.3.2 Evaluation Results & Data Analysis

Basic statistics. In Table 7.1, we summarize the details of the datasets obtained from the first and second evaluation studies, namely, Lab and Crowd datasets, respectively. As expected, the Crowd dataset is not only larger but also much more diverse with respect to age and gender. We also observed considerable diversity for the country and education level of the participants in the latter dataset, as shown in Tables 7.2 and 7.3, respectively.

In Table 7.4, we provide the number of the annotated posts for each year from 2009 to 2014. For earlier years of 2007 and 2008 we don't have a large enough number

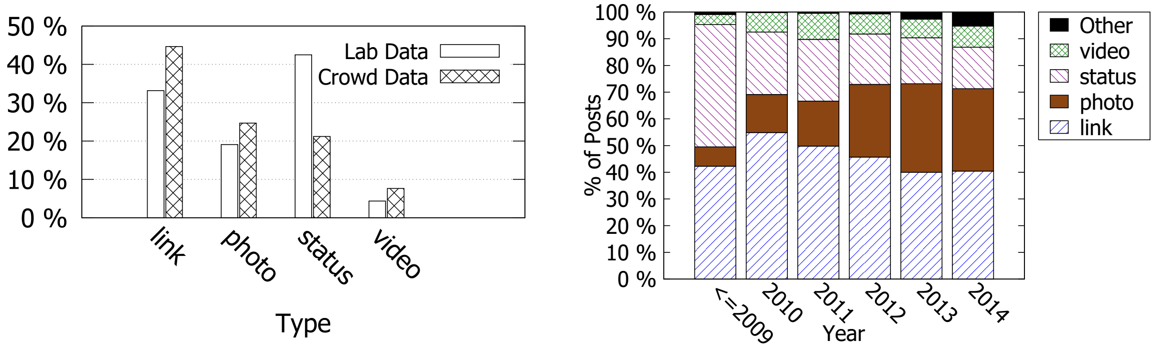
Table 7.4. Number and percentage of the evaluated posts per year.

Lab Dataset	**Year**	**No. of Posts**	**Percentage**
	≤ 2009	1140	13.42
	2010	1,367	16.09
	2011	724	8.52
	2012	1,657	19.51
	2013	3,303	38.89
	2014	303	3.57
Crowd Dataset	**Year**	**No. of Posts**	**Percentage**
	≤ 2009	3,514	6.13
	2010	7,571	13.22
	2011	10,840	18.92
	2012	10,425	18.20
	2013	13,635	23.80
	2014	11,296	19.72

of posts and hence, we aggregate them with those from 2009. We can observe that in the Lab dataset, there is an imbalance in the distribution of data annotated from each year, as the percentage of posts annotated per year varies from about 3% to 38% between 2009 and 2014. In contrast, our Crowd dataset seems to be more stable in this sense, especially between the years 2011 to 2014 (as the percentages are in the range of 18% to 23% for all years in this period).

At the time of our evaluation, Facebook had seven types of posts (namely, link, checkin, offer, photo, question, swf and video) that basically describes the type of content attached to a post. In Figure 7.1(a), we present the distribution of these types among the evaluated posts in our studies. In the Lab dataset, the most popular post type is status update (42.5%) followed by shared links (33.1%), photos (19%) and videos (4%). The second dataset, Crowd, has a slightly different distribution where posts of type shared link (44.4%) is the most popular and followed by photos (24.7%), status updates (21.1%) and videos (7%). Note that, in both datasets we disregard the other post types that are infrequent (i.e., less than 1%).

We also investigated the distribution of different post types over years, presented in Figure 7.1(b) for the Crowd dataset. Our observation is that there is a clear increase in the use of photos and videos over time. The number of photos increased from 7% in 2009 to about 30% in 2014. For video we have an increase from 3% to 7% in 2014. These numbers are taken from our larger Crowd dataset, but we can observe a similar trend in the Lab dataset. Several factors help us to explain this change in behavior. First, the catch up of broadband connection allowed users to quickly upload large amounts of data (photos and videos). Second, the dissemination of smart phones with embedded cameras played an important role. Nowadays, anyone can quickly take a snapshot and upload it on the Web. Statistics from photo sharing



(a) (b)

Figure 7.1. (a) Percentage of post types for each dataset, (b) Percentage of post types per year for the Crowd dataset.

website Flickr[6] show that the most used cameras are, by far, embedded smart phone cameras[7]. The rate of links and status information changes over years, however, there is no clear trend seen.

Analysis of Evaluation Results. In this section, we present an analysis of the evaluation results and focus on a set of promising features (from the categories of social, temporal and network features, as will be described in the next section) that are most likely to be useful for identifying memorable posts.

We first take a look at the overall distribution of ratings in our datasets, shown in Figure 7.2. We observe that in both datasets the portion of posts with rating 0 dominates with 57% for the Lab dataset and 37% for the Crowd dataset. In contrast, the fraction of posts that are given the highest rating is only 6% and 21% in the Lab and Crowd datasets, respectively. This indicates that participants consider a significant fraction of their posts worthless to retain for future, and justifies our work that aims to characterize this relatively small portion of posts, which are memorable, and generate rankings to present such posts at top.

We also analyzed the distribution of ratings wrt. the post types. We find that posts of type *photo* have the highest average rating, namely 1.93 and 3.10 for the Lab and Crowd dataset, respectively. In both datasets, *video* is the type with the second highest average rating (i.e., 1.27 for the Lab and 2.78 for the Crowd dataset). The average ratings of types *status update* and *link* are found to be considerably lower (especially for the Lab dataset), suggesting that posts with type *photo* or *video* are more likely to be memorable.

Content age for retention. Next, we focus on the role of time in deciding on content retention, i.e., whether older content on the average is rated lower than more recent content. For this purpose, we investigate the relationship between the post

[6] http://www.flickr.com

[7] http://www.flickr.com/cameras

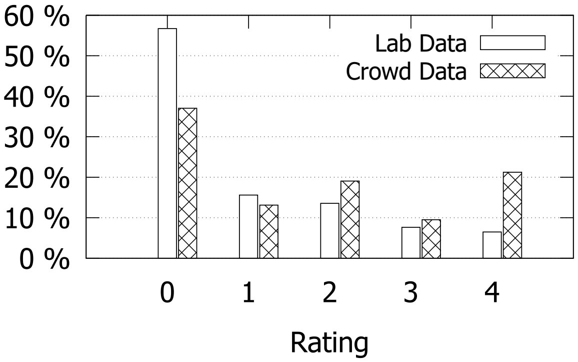


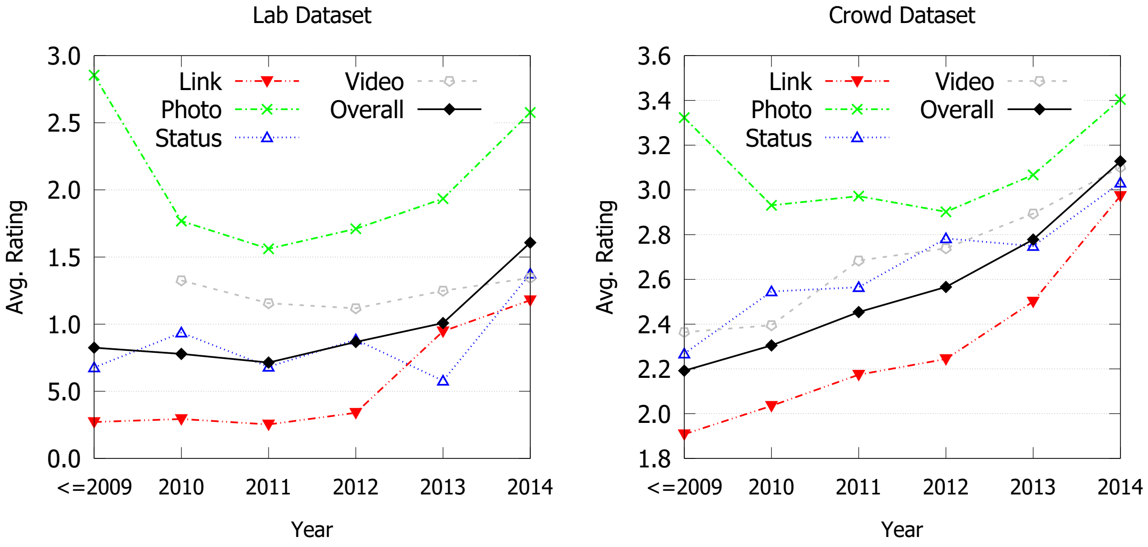
Figure 7.2. Distribution of the user ratings for each dataset. The average of ratings is 0.92 for the Lab dataset and 1.65 for the Crowd dataset.

ratings and age of post. In Figure 7.3, the solid line shows the average rating for the different years of content creation. The figure reveals a clear trend where participants in the evaluation assigned higher ratings to more recent posts. This is in line with the idea of a decay function (as widely used in the field of data streams [CS06, PVK$^+$04]) underlying the content retention model. The decrease in the average rating with growing age of content is especially steep in the first years (2013 and 2014). Surprisingly, we also see an increase in the rating values for the year 2009 in the figure, which we attempt to explain using a fine-grain analysis of ratings, i.e., per post type, in the following paragraph.

In Figure 7.3, we see the trend for the average ratings for individual post types denoted with the dashed lines[8]. Once more, we observe an increase of ratings for the most recent posts. However, we also see very high average ratings for the oldest photos (older than 5 years). Thus, we conjecture that seeing these older (already forgotten) photos again caused some positive surprise for the users, which resulted in higher ratings. Indeed, this perception would also support the idea of creating Facebook summaries for reminiscence, yet we leave its verification (maybe via face-to-face participant interviews) as a future work. Note that the same trend (of rating more recent content as more worthy to retain) holds for both datasets, yet as shown in the figure, the Crowd dataset is exhibiting a smoother behaviour for different post types.

Finally, Figure 7.4 demonstrates the same trend from a different perspective (given for only Crowd dataset for brevity). In this figure, the black line indicates the total percentage of posts considered as memorable (i.e., those with a rating greater than 0), which increase consistently over the years, while the red line shows the the percentage of posts rated with 0, exhibiting an opposite trend. Overall, these findings suggest that content age may serve as an important feature to identify memorable posts.

[8] For the Lab dataset, videos are shown for year 2010 and afterwards, as the number of videos before 2010 is very small in this dataset.



(a) (b)

Figure 7.3. Average rating of all posts per creation year (the solid black line) and average rating of posts for each content type per creation year (dashed lines).

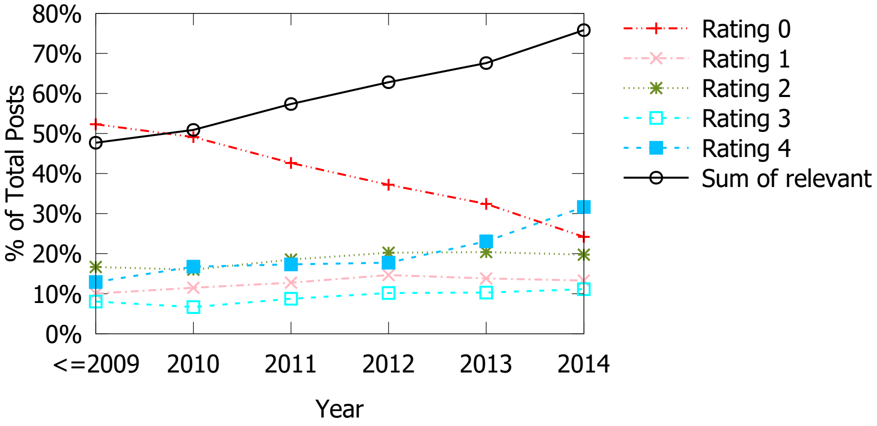


Figure 7.4. Percentage of posts per rating for each year. The black line denotes the percentage of all the posts with a rating greater than 0 for each year (for the Crowd Dataset).

Number of likes and comments for retention. On Facebook, it is possible to *comment* for or *like* a particular post, as common forms of expressing community feedback. In our larger Crowd dataset, 70% (80%) of the posts lack any likes (comments), while 26% (18%) of the posts have between 1 to 10 likes (comments), respectively.

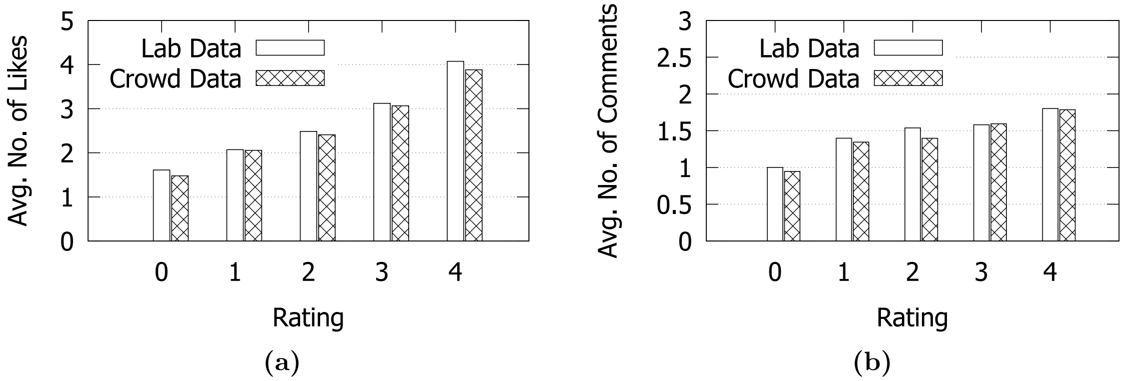


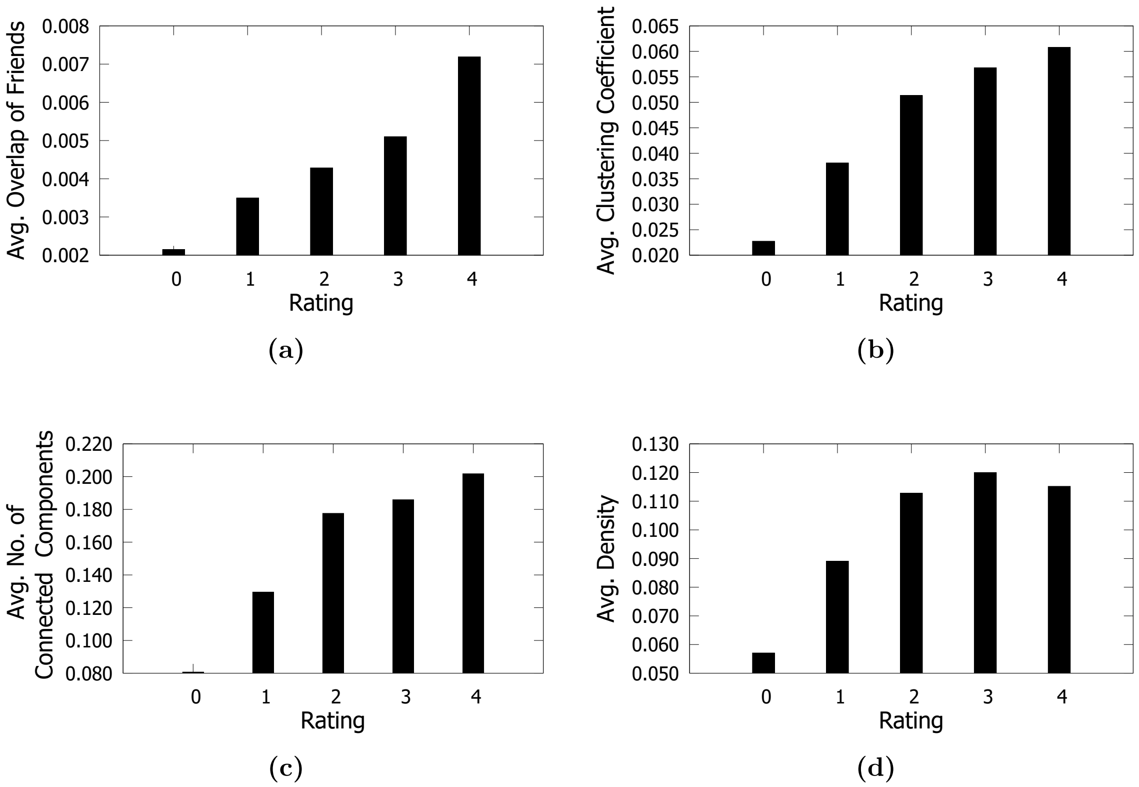
Figure 7.5. Avg. number of (a) likes, and (b) comments for the posts per rating.

Figure 7.5 reveals that for the posts with higher ratings, the average number of likes (comments), is also higher. This trend holds for both datasets, and also confirms our preliminary findings in [NKKN14] that involved a smaller number of participants than those of the studies reported here. This indicates the robustness of this observation. Thus, the number of likes and comments seem to be among the crucial features to characterize the memorable posts.

Network features for retention. To better understand the importance of connections of the users involved (e.g., liked, commented, or tagged) in a post, we analyse for each post a set of network measures capturing two main effects. First, the relationship between the users' social graph and the users' involved in a post. Here, our assumption is that posts involving more people from the user's friendship graph may have a higher probability of being relevant for retention than other posts with a few friends in their social graphs. To this end, we compute the feature *overlap of friends*, which is the ratio of the friends of a user to all people who are involved in a post. Secondly, we are interested in the relationships within the social graph of each post to identify differences in their users connections. In this case, our assumption is that a high connectivity within the users involved in a post can lead to a higher chance that a post is considered relevant for future needs. To this end, we capture the graph connectivity by standard network measures, such as the *clustering coefficient* [WS98], *number of connected components* [Tar72], and *density* [CM83].

In Figure 7.6, we present the average values for these four network features over the posts for each rating (for the Crowd dataset). While computing these features for a given post, we only considered the users who *liked* the post, i.e., the *likes-network*. The figure shows that posts with higher ratings exhibit higher scores for these features, implying that such features may also be useful in identifying memorable posts.

Summary. From the previous statistics and analysis, we can deduce first ideas for determining features that have a high impact in the identification of memorable posts.



(a)

(b)

(c)

(d)

Figure 7.6. Average feature score computed over the network of users who liked a post vs. post rating, for the network features (a) overlap of friends, (b) clustering coefficient, (c) no. of connected components, and (d) density.

Roughly speaking, recent photos and videos with high number of likes/comments and high overlap/connectivity within their social graphs of likes seem to be the best candidates for retention.

7.4 Candidate Features for Retention

While our data analysis presented in the previous section allows us to detect a small set of seed features, machine-learning based approaches for tasks similar to ours typically employ all potentially useful features (e.g., thousands of features are used for training ranking models for search engines [YHT+16]) that can be extracted from the data, as a feature that is found to be less useful on its own can improve the overall performance of the model when combined with other features. Therefore, for capturing factors that might influence retention decisions of users, we compiled a broad set of 111 features. They can be categorized into five groups described as follows, while each feature is individually described in Table 7.5:

- Temporal features: The inclusion of temporal features is inspired by the idea that retention preferences are influenced by a decay function as it was also confirmed by the data analysis in the previous section. For temporal features, we consider the temporal aspect of the post in terms of creation date, age, and lifetime. While *age* is the time between the evaluation and creation date, i.e., the time the post was created, *lifetime* is measuring the active time of a post starting at the time it was created to the last update. We also use variants of the age feature, which use the time of the last update and the time of the last commenting, respectively, instead of the creation time.

- Social features: The social features capture core signals of social interaction in a Social Web application, covering the features that are typically used in Facebook analysis: *number of likes*, *number of comments*, and *number of shares*.

- Content-based features: We use the *type* of posts as well as some specific features extracted from the metadata of the post (as provided by Facebook) such as the *status type*, textithasLink, *hasIcon*, and *app type*. To respect user privacy, the only text-based feature in our set is the length of text included in posts and comments. In other words, we do not utilize the textual content of the posts.

- Privacy features: These are based on the privacy settings for a post that are specified by its owner to restrict the access of this post to a particular set of user.

- Network features: Based on our analysis in the previous section, for each post we extract seven different network feature as presented in 7.5. We compute these features from three different graphs for each post, namely, the graph of users who liked the post, graph of users who commented on the post, and graph of all users who liked, commented or tagged in the post. We employ the implementations of these features as provided by the Gephi project [9] [10].

We also apply a personalized normalization to the social and network features to capture the individual characteristics and behavior of users more accurately. Furthermore, each categorical feature (like *type*) is mapped to multiple binary features (e.g., *type_IsLink*, *type_IsPhoto*, *type_IsStatus*, etc.). After these normalization and binarization steps, we end up with 111 features including 5 temporal, 8 social, 39 content-based, 13 privacy, and 46 network features. In Table 7.5, we provide a brief description for each feature.

Table 7.5. The list of features extracted for each post.

Feature	**Category**	**Description**
No. of Likes	Social	No. of people who like this post.
No. of Comments	Social	No. of comments on the post.
No. of Shares	Social	No. of shares of the post.
No. of Tagged Users	Social	No. of users mentioned in a post.
No. of Likes on Comments	Social	Total no. of likes included in the comments.
Created-Time	Temporal	The creation time of the post.
Lifetime	Temporal	The time between creation and last update of the post.
Age (Creation time)	Temporal	The age of the post between the day of the evaluation and the time the post is created, updated or commented last time.
Age (Update time)	Temporal	
Age (Last comment)	Temporal	
Privacy Settings	Privacy	A privacy class setting for access to the post, e.g. *everyone*, *friends_of_friends*, *all_friends*, *custom*, *self*, *null*
No. of Users (Allowed)	Privacy	No. of the specific users or friends (in lists) who can see the post.

[9] https://github.com/gephi
[10] https://gephi.org/

Table 7.5. The list of features extracted for each post.

Feature	Category	Description
No. of Users (Denied)	Privacy	No. of the specific users or friends (in lists) who are not allowed to see the post.
Privacy Friends	Privacy	No.of users (in a customized category) who can see the post, e.g. *some_friends*
hasDescription	Privacy	Post has a description of the privacy settings.
Type of Post	Content-based	Type of a post with values such as *link*, *status*, *photo*, *video*, etc.
Status Type of Post	Content-based	Description of the type of a status update. Values are *added_photos*, *added_video*, *created_group*, etc.
hasMessage	Content-based	The post contain a status message.
hasStory	Content-based	Text from stories that are not intentionally generated by users, e.g., when someone else posts on the person's profile.
hasDescription	Content-based	A description to a particular content, e.g., a website.
hasLink	Content-based	A link is attached to the content.
hasCaption	Content-based	The caption of a link is in the post.
hasIcon	Content-based	The post has a link to an icon representing the type of this post.
Length of Message	Content-based	The length of the corresponding textual section in the post.
Length of Story	Content-based	
Length of Description	Content-based	
Length of Comments	Content-based	Total length of the comments for the post.
Send by Mobile-APP	Content-based	Post includes information about the app that was used to publish it.

Table 7.5. The list of features extracted for each post.

Feature	Category	Description
Type of APP	Content-based	The type of the app the post was published by. Values are *mobile*, *others*, and *null*.
Self Posted	Content-based	Content is posted by the user herself.
Self Liked	Content-based	The post is liked by the user herself.
Self Commented	Content-based	The user commented on her post.
Overlap of Friends	Network	The ratio of the user's friends involved in the social graph created for a post.
Clustering Coefficient	Network	The clustering coefficient of the social graph created for a post.
Degree	network	The degree of the social graph created for a post.
Connected Components	Network	The no. of connected components in the social graph created for a post.
Density	Network	The density of the social graph created for a post. A complete graph has all possible edges and density equal to 1.
Diameter	Network	The diameter is the maximal distance in the social graph created for a post.

7.5 Ranking Posts for Retention

Based on the candidate features described in the previous section, our goal here is ranking a user's posts to identify the most memorable ones. To this end, we adopt strategies from web search domain, where machine-learned rankers are heavily investigated and incorporated into commercial search engines (e.g., see [YHT+16]). If we make an analogy, a user in our case corresponds to a query in the search setup, and user's posts correspond to the documents retrieved for the query. During the training stage, for a given user u, we construct an $m-$dimensional feature vector F for each post of this user, and augment the vector with the rating r assigned to this post in the evaluation study. For the testing, we feed vectors in the form of (u, F) to the learnt model for each user in the test set; and the model outputs a ranked list of posts for each user. We evaluate the success of the models using a typical metric from the literature, namely, Normalized Discounted Cumulative Gain (nDCG), which is a rank sensitive metric that takes into account graded labels. We report nDCG scores at the cut-off values of {5, 10, 15, 20}. In the following sections we consider general and personalized ranking models based on this settings, where the personalized models aim to take into account differences in characteristics and preferences of users with respect to retention.

7.5.1 General Ranking Models with Feature Selection

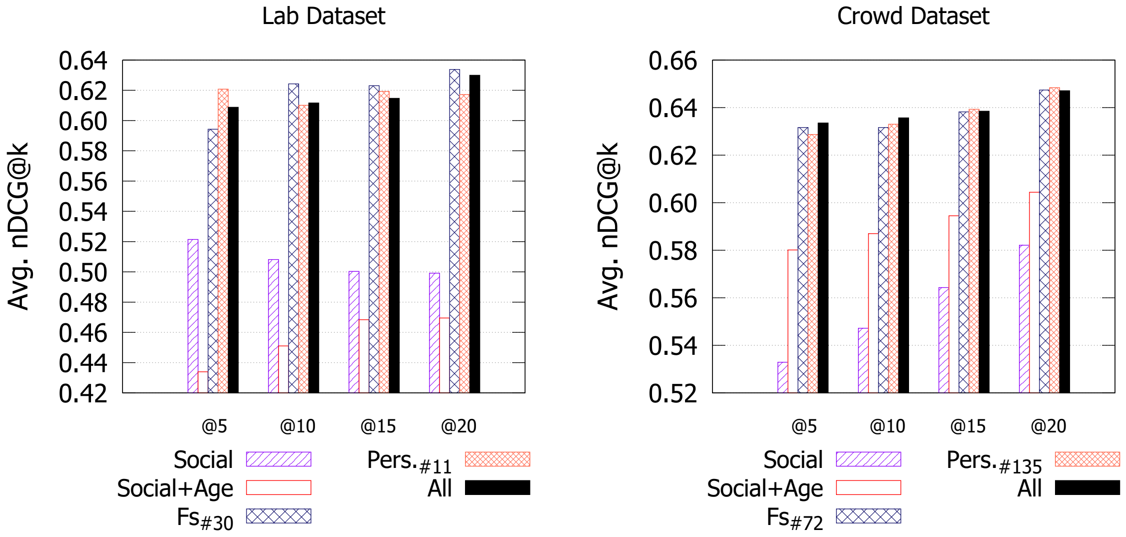
In the experiments, we employ a well-known algorithm, namely RankSVM, from learning-to-rank literature [Joa02b]. Instead of single data instances, RankSVM considers the pairs of instances (posts of a user, in our case) while building a model. We apply leave-one-out cross validation for both of our datasets. For the Lab dataset, we use all 8,494 (posts) from 41 users, as described before. For the larger Crowd dataset, we randomly took 100 posts per user to avoid class imbalance (as there were some users who evaluated much more than 100 posts), which resulted in 47,000 posts for 470 users.

To the best of our knowledge, we are the first to propose ranking social media posts for retention, hence, in the literature, there does not exist a baseline set of features that is specified for our task. Therefore, we train two intuitive baseline models taking into account our findings on features for retention from our data analysis presented in Section 3, taking a practical approach.

In the first baseline, `Social`, we use basic social features, namely, the *number of likes*, *number of comments* and *number of shares* (and their versions normalized per user). We choose the latter features as they are the most intuitive popularity signals in social web and hence, likely to be involved in practical applications, such as the Facebooks apps discussed before[11]. Our data analysis has also yielded evidence that number of likes and comments can be useful for identifying memorable posts. For the second baseline, `Social+Age`, in addition to social features we use a temporal feature, *age* (wrt. the creation time), to build our models, as this feature is again found very promising in Section 3.

Figure 7.7 reveals the performance of RankSVM for ranking posts using all the proposed features for the Lab and Crowd datasets. As a first observation, we see that the baseline models differ in performance for the two datasets. The `Social` baseline performs better for the Lab while `Social+Age` baseline performs better for the Crowd dataset. This might be due to the observation that, as shown in Figure 7.3, the relationship between content age and rating is weaker for the Lab dataset than that for the Crowd dataset. Nevertheless, in the following, all the expressions claiming an improvement over a baseline refers to the baseline that performs better for the

[11] For instance, while Facebook's "Year in Review" does not disclose how the content for each user is tailored, it is stated that the *number of mentions* in the posts is used to determine the top-10 topic list for the platform itself.



(a) (b)

Figure 7.7. Effectiveness of the ranking models for (a) Lab, and (b) Crowd dataset. `Social` and `Social+Age` denote the baselines, `All` denotes the general ranking model with all features, FS_X denotes the general model with X features (after feature selection) and $Pers._K$ denotes the personalized model using K nearest neighbors of each user.

dataset in question.

Our results presented in Figure 7.7 further show that the candidate features presented in Section 7.5 are actually very useful, and using all these features (denoted as `All`) for training a ranker yields relative effectiveness improvements of up to 9.21% (from an nDCG@5 score of 0.58 to 0.63) and 16.8% (from 0.52 to 0.61) over the baselines, for the Lab and the Crowd dataset, respectively. For the latter set, relative improvements in nDCG scores are even larger for the higher cut-off values of 10, 15 and 20; being 20.4%, 22.9% and 26.2%, respectively.

Apart from the relative improvements over the intuitive baselines, we believe that the range of the effectiveness scores for our general models (i.e., an nDCG score of up to 0.64) is reasonable in comparison to state-of-the-art performance in typical learning-to-rank settings optimized for relevance. For instance, a recent work reports that over Microsoft and Yahoo challenge datasets (each with around 30K queries), a state-of-the-art ranker yields nDCG@10 scores of 0.49 and 0.78, respectively [GLNP16]. For ranking tweets, an approach again with RankSVM is shown to yield nDCG@10 scores less than 0.60 [DJQ+10]. This indicates that our approach in this chapter, i.e., training models to rank social media posts for retention, is appropriate and effective.

The next question we address is: Can we identify a subset of the candidate features that has the highest impact in ranking memorable posts? While feature selection methods are widely applied for various classification tasks, only a few works have investigated their performance in a learning-to-rank framework [GLQL07, DC10, NA14, GLNP16]. Here, we adopt the so-called GAS (Greedy search Algorithm of Feature Selection) introduced by Geng et al. ([GLQL07]). In GAS, we compute each feature's isolated effectiveness, but additionally, we also compute pairwise feature similarity, i.e., to what extent the top-20 rankings generated by two different features correlate. To compute the similarity of two ranked lists, we use Kendall's Tau metric. Then, the feature selection proceeds in a greedy manner as follows: In each iteration, first the feature with the highest effectiveness score is selected. Next, all other features' effectiveness scores are discounted with their similarity to the already selected feature. The algorithm stops when it reaches the required number of features, N. We experiment for all possible values of N, from 1 to 111 (as $N = 111$ is the case with all features), and evaluate the performance. Figure 7.7 also shows the results for the feature selection strategy with the best-performing value of N, which is found to be 30 (27% of all features) and 72 (64.9%) for the Lab and the Crowd datasets, respectively. Remarkably, although they are trained with a subset of all features, these smaller models still yield comparable (and sometimes, slightly better) effectiveness wrt. the models using all features, especially for the Crowd dataset.

For this latter experiment, we analyze the features selected by GAS in each fold (recall that we have leave-one-out cross validation) to identify the most promising features for the task of ranking posts. As the absolute value of the weights assigned to features by the RankSVM model (built using a linear kernel) can reflect the importance of features [CL08], we averaged these absolute values for the features appearing in the model learnt for each fold. Then, we determined top-25 features with the highest average scores in the models built after feature selection with GAS, for our Lab and Crowd datasets, separately.

In Table 7.6, we present the common features that appear among the top-25 features of both datasets (the average rank column denotes the position of the feature in top-25 list for a given dataset). We observe that these 16 features fall into four of the categories described before, while no features from the privacy category could get into the list. It turns out that temporal features (along with their variants) and basic social features (no. of likes and comments) are among the most effective for the ranking models. There are network features computed over all the users involved in the post (i.e., those who liked, commented or tagged), as well as content-based features, namely, type and length of the post. This list verifies our analysis presented in Section 3, and further demonstrates that it is helpful to have various variants of the same feature (e.g., normalized or computed in alternative ways), as a learning algorithm can benefit from all. Finally, some features (like the content length) that may not seem to be promising on its own at a first glance turn out to be useful when used in combination with others.

Table 7.6. The common features in the top-25 features computed for Lab and Crowd datasets (along with the feature's rank in each list).

Category	**Feature**	**Rank in Lab Dataset**	**Rank in Crowd Dataset**
temporal	Age (created time)	1	7
temporal	Created Time	2	2
temporal	Age (last Updated Time)	3	16
temporal	Lifetime	4	12
temporal	Age (last comment)	5	15
social	No. of likes	9	17
social	No. of Comments	12	20
social	Pers. No. of Likes	24	22
network	Overlap. No. of Friends (all)	16	9
network	Density (all)	17	18
network	Pers. Density (all)	7	14
content-base	Type	6	21
content-base	Pers. Length Message	8	4
content-base	Length Story	10	5
content-base	Pers. Length Story	15	3
content-base	Length Description	22	23

7.5.2 Personalized Ranking Models

So far, we considered a general ranking model learnt for all the users. However, in search domain, recent studies have shown that it is beneficial to build query-dependent ranking models, as queries significantly differ from each other (e.g., [GLQ+08, CCM14, ZHLL12]). In particular, Geng et al. ([GLQ+08]) propose to use k-Nearest Neighbor (kNN) method so that for a given query first its nearest neighbors are found in the training set and then a customized ranker is learnt using only these neighbor instances. Analogously, in our setup, it is natural to hypothesize that similar users may have similar motivations and preferences while deciding on the memorable posts. Hence, we also apply a kNN based strategy to build more personalized ranking models.

We represent each user with a vector of three key features, namely, the *number of posts*, *number of friends*, and *number of connections* among the user's friends, which may reflect the coherence in the user's network. We anticipate that these user centric features best capture the activity level of a user in a social media application, and users with similar activity patterns can exhibit similar behavior while deciding on the memorable posts. To determine the nearest neighbors of a user, we compute the Euclidean distance between the pairs of these feature vectors, and choose the ones (k of them) that yield the smallest distances. Then, for each test user, only these k nearest neighbors (and their posts) are used to train the RankSVM algorithm.

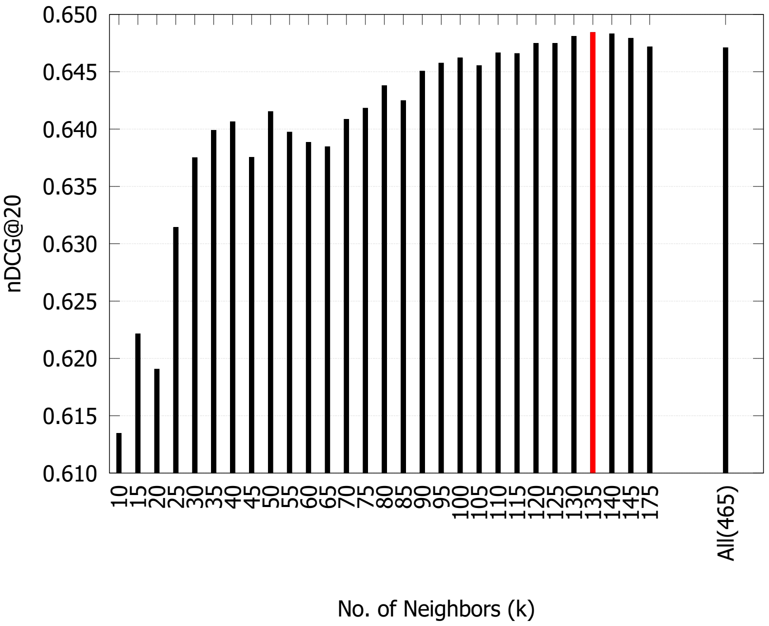


Figure 7.8. Effectiveness of the personalized ranking model vs. number of neighbors, k, for kNN (for the Crowd dataset).

In Figure 7.8, we present the performance of personalized models vs. k, the number of neighbors in kNN, which is in the range [1, 469] for the Crowd dataset (the trend for the Lab dataset is similar and not shown here for brevity). Note that, the figure does not include k values greater than 175, as the effectiveness score does not vary much after this point. The figure shows that training with very small number of neighbors (e.g., less than 25) may cause losses in the model effectiveness, and the best results are obtained for k =135.

In Figure 7.7, we also present the performance for personalized ranking of posts[12]. As the best results are obtained when we set the number of neighbors k to 11 for the Lab dataset and to 135 for the Crowd dataset, we report only these cases. Our results are encouraging in that, for both datasets, the personalized approach can provide gains in comparison to using a general ranking model for various cut-off values (cf., compare the fourth and last bars in Figure 7.7 for each cut-off value). Most remarkably, for the Crowd dataset, while nDCG@5 score is 0.608 for the general model using all features, personalized model achieves a score of 0.620, providing a relative improvement of about 2%. We envision that in a real setup where millions of users exist with different habits of interacting with the social applications, the idea of building ranking models customized for individual users might improve the effectiveness even more.

Finally, we also experimented with feature selection in the personalized setup, where we applied the GAS strategy for the k nearest neighbors of each user. It turns out that, feature selection diminishes the benefits obtained by building personalized rankers and hence, we do not provide results for this experiment. Given that the training is already restricted to a small set of neighbors, we conclude that it may not pay off to apply feature selection when we aim to build specific models per user.

Note that, a final concern for building personalized ranking models could be efficiency. In the case of the web search, training a model for each query can imply prohibitive online processing costs, as the users typically expect search results in less than a second [GLQ+08]. However, in our case, this would be less of a concern; as ranking the posts for retention is not an everyday task for a user, but an application that is most likely to be executed periodically, such as de-fragmenting your hard-drive. Hence, the additional processing latency for online model building can be tolerated by the users, for the promise of a better final ranking. Furthermore, it is still possible to improve the efficiency using offline pre-processing techniques, such as clustering, as proposed in an earlier work [GLQ+08]. Thus, both from the effectiveness and efficiency perspectives, we conclude that building personalized ranking models for retention arises as a promising direction.

Summary. Our experiments presented in this section show that general models trained with 111 candidate features yield reasonable effectiveness (nDCG scores over 0.61 for all cut-off values and datasets) and outperform intuitive baselines (using social and temporal features) with a large margin (up to 26%) for ranking posts

[12] While we regret to make the reader refer to back to check this figure, we preferred to present the performance of all ranking models in a single figure for the sake of comparability and brevity.

for retention. We also demonstrated that these general models can be made more compact by feature selection, and even after this, the performance is comparable to the models using all the features. Finally, we built personalized ranking models that can provide a relative improvement of about 2% over the general models.

7.6 Conclusions

In this article, we lay the foundations towards the creation of life summaries from a social media platform, Facebook. This is a non-trivial challenge that requires accurate ranking of memorable posts in a user's timeline. In order to address this challenge, one first needs to assess users' perception of what is important for retention in a social platform. To this end, we conducted two user evaluation studies: The first study involved 41 participants from the research communities and yielded 8,494 annotated posts, while the second study involved 470 participants recruited from a crowdsourcing platform and yielded 57,281 annotated posts.

On this invaluable corpus, we conducted a primary data analysis and identified a small set of seed features that are most likely to characterize memorable posts. Next, leveraging a broader set of candidate features extracted for each annotated post, we trained both general and personalized models to rank the posts. These rankers are effective, as they can outperform a practical baseline that employ the most intuitive features identified during our data analysis, and as they yield effectiveness scores comparable to the recent works that again employ machine-learnt ranking models for a different yet related purpose, namely, traditional document retrieval. A question that still remains open for exploration is whether it is possible to further increase the effectiveness of the rankers by taking into account the textual content of the posts, which lies in a grey area involving hot debates on user privacy issues.

In our experiments, by applying various feature selection techniques, we could identify a compact set of features that captures the most discriminative representatives of different feature categories as we define here (namely, content-based, temporal, social, network, and privacy), and yield ranking models that are as effective as those with all the features. This is also valuable, not only for building models more efficiently in large scale systems, but also for figuring out the directions we need to concentrate in future user studies for a more fine-grained understanding of the human retention preferences in social media applications.

In our future work, we plan to address grouping of related posts of a user for structuring the information space and develop effective ways of generating concise and diverse summaries over such groups of posts for retention.

8 Conclusion and Future Work

In this thesis, we presented studies and approaches for improving access, sharing, and management of information in modern web applications such as Web search engines and online social networks. In this final chapter, we conclude the contributions and discuss possible future work.

Diversification for improving information access. Information access brings various challenges for large-scale retrieval systems such as modern web search engines. In Chapter 3, we addressed approaches for web search result diversification, which can improve retrieval effectiveness for ambiguous queries. In particular, we presented two greedy algorithms for improving the efficiency of an implicit diversification algorithm without compromising effectiveness. Our proposed algorithms can reduce the computational cost up to 97%, which makes them applicable for online processing. Further, in an depth-study we investigated the behavior of implicit and explicit diversification algorithms on a large-scale setup with distributed nodes. Our findings show that diversification on the broker yields often to better result quality than diversification on the nodes with an acceptable cost for network communication. In Chapter 4, we addressed the problem of feature selection for learning-to-rank. Here, we presented three diversification algorithms exploited for feature selection that can outperform baselines from the literature on standard datasets.

Future Work. Further, we pointed out that reducing the number of features in a real setup can improve computational cost significantly. Future research can focus on the distributed diversification for other scenarios and investigate approaches to further reduce the network communication costs when diversification is applied at the broker. Further, other diversification methods can be exploited and evaluated for feature selection for improving retrieval effectiveness.

Privacy issues and communication practices through information sharing. In Chapter 5 and Chapter 6, we investigated privacy and communication practices in social and professional networks. In Chapter 5, we investigated the usage of privacy settings in online social networks and presented an approach for supporting users in selecting adequate privacy settings for their posts before sharing them to a wider audience. Our analysis shows that users show clear difference in their usage of privacy settings, and that the type of post has a significant impact on the selected privacy setting by the user, e.g. "photo" and "video" are more often restricted to a smaller audience. Further, we proposed a set of features from different categories that can be used for predicting the privacy settings of posts. Our experiments show that our general model can predict the privacy of posts with a recall and precision of 80%.

In Chapter 6, we presented an in-depth study on the communications practices and behavior of users within different platforms, in particular Facebook, Twitter, and LinkedIn. Our research show that users with different skills have specific rules and intentions for using a particular social media. The insights presented suggest separated analysis and treatment of activations with different skills networks and subgroup which can support the users to find specific groups for people and resources. Our privacy analysis demonstrated clear differences in users' behavior with respect to privacy settings which indicate personalized model for privacy prediction might further improve the prediction of highly private posts.

Future Work. The insights of our cross platform study on communication practices suggest a further analysis on treatment of activities within subgroups, and investigating new supportive functionalities in professional networks to support the user needs.

Personalized and long-term information management for social content. In Chapter 7, we addressed the problem information management of identifying social content for retention. In our study, we conducted two evaluation studies with 41 and 470 participants which annotated 57,281 posts with respect to their importance for retention on Facebook. We presented an in-depth analysis of features which could characterize memorable posts. In addition to that, we trained both general and personalized models to rank the posts. Our proposed approach yield to higher effectiveness score comparable to the baseline. In our experiments, we presented a set of top features extracted using various feature selection methods.

Future Work. Our results presented in Chapter 7 are valuable for building models in large scale systems. A possible future direction would be further investigate another features for retention such as textual feature. Further, grouping of related posts of a user for structuring the information space and develop effective ways of generating concise and diverse summaries over such groups of posts for retention.

Bibliography

[AGHI09] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the 2nd International Conference on Web Search and Web Data Mining*, pages 5–14, 2009.

[AHH^{+}13] Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, and Daniel Krause. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2-3):169–209, 2013.

[AHK11] Fabian Abel, Eelco Herder, and Daniel Krause. Extraction of professional interests from social web profiles. *Proc. Augmented User Modeling at UMAP*, 34, 2011.

[BdR06] Krisztian Balog and Maarten de Rijke. Finding experts and their details in e-mail corpora. In *Proceedings of the 15th international conference on World Wide Web*, pages 1035–1036. ACM, 2006.

[BHJ09] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. In *ICWSM*, pages 361–362, 2009.

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[BP11] Simon Bowen and Daniela Petrelli. Remembering today tomorrow: Exploring the human-centred design of digital mementos. *International Journal of Human-Computer Studies*, 69(5):324–337, May 2011.

[Bre96] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Bre12] Brian Brett. The psychology of sharing, 2012. Available at http://nytmarketing.whsites.net/mediakit/pos/.

[BRL07] Christopher J. Burges, Robert Ragno, and Quoc V. Le. Learning to rank with nonsmooth cost functions. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 193–200. MIT Press, 2007.

[BYRN99] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[Car09] Ben Carterette. An analysis of NP-completeness in novelty and diversity ranking. In *Proceedings of the 2nd International Conference on the Theory of Information Retrieval*, pages 200–211, 2009.

[CASN13] Sergiu Chelaru, Ismail Sengor Altingovde, Stefan Siersdorfer, and Wolfgang Nejdl. Analyzing, detecting, and exploiting sentiment in web queries. *ACM Trans. Web*, 8(1):6:1–6:28, December 2013.

[CBY11] BerkantBarla Cambazoglu and Ricardo Baeza-Yates. Scalability challenges in web search engines. In Massimo Melucci and Ricardo Baeza-Yates, editors, *Advanced Topics in Information Retrieval*, volume 33 of *The Information Retrieval Series*, pages 27–50. 2011.

[CC00] Padraig Cunningham and John Carney. Diversity versus quality in classification ensembles based on feature selection. In *ECML 2000*, pages 109–116, 2000.

[CC09] Ben Carterette and Praveen Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM Conf. on Information and Knowledge Management*, pages 1287–1296, 2009.

[CC11] O. Chapelle and Y. Chang. Yahoo! learning to rank challenge overview. In Olivier Chapelle, Yi Chang, and Tie-Yan Liu, editors, *Proceedings of the Learning to Rank Challenge*, volume 14 of *Proceedings of Machine Learning Research*, pages 1–24, Haifa, Israel, 25 Jun 2011. PMLR.

[CCM14] Ethem F. Can, W. Bruce Croft, and R. Manmatha. Incorporating query-specific feedback into learning-to-rank models. In *Proc. of SIGIR '14*, pages 1035–1038, 2014.

[CCNP06] Paul-Alexandru Chirita, Stefania Costache, Wolfgang Nejdl, and Raluca Paiu. Beagle^{++}: Semantically enhanced searching and ranking on the desktop. In *Proceedings of the 3rd European Semantic Web Conference*, ESWC '06, pages 348–362, 2006.

[CCS09] Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. Overview of the trec 2009 web track. In *Proceedings of the 18th Text Retrieval Conference*, 2009.

[CCSC10] Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Gordon V. Cormack. Overview of the trec 2010 web track. In *Proceedings of the 19th Text Retrieval Conference*, 2010.

[CDR12] Claudio Carpineto, Massimiliano D'Amico, and Giovanni Romano. Evaluating subtopic retrieval methods: Clustering versus diversification of search results. *Inf. Process. Manage.*, 48(2):358–373, 2012.

[CG98a] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM.

[CG98b] Jaime G. Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336, 1998.

[CGGG17] Abhijnan Chakraborty, Saptarshi Ghosh, Niloy Ganguly, and Krishna P. Gummadi. Optimizing the recency-relevancy trade-off in online news recommendations. In *In Proc. of WWW '17*, pages 837–846, 2017.

[CH12] A. Coman and W. Hirst. Cognition through a social network: the propagation of induced forgetting and practice effects. *J Exp Psychol Gen*, 141(2):321–36, 2012.

[Che05] Ying Chen. Information valuation for information lifecycle management. In *Proceedings of International Conference on Autonomic Computing*, 2005.

[CHNS14] Sergiu Chelaru, Eelco Herder, Kaweh Djafari Naini, and Patrick Siehndel. Recognizing skill networks and their specific communication and connection practices. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, HT '14, pages 13–23, New York, NY, USA, 2014. ACM.

[CJL+11] Olivier Chapelle, Shihao Ji, Ciya Liao, Emre Velipasaoglu, Larry Lai, and Su-Lin Wu. Intent-based diversification of web search results: metrics and algorithms. *Inf. Retr.*, 14(6):572–592, 2011.

[CK06] Harr Chen and David R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 429–436, 2006.

[CKC+08a] 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 International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–666, 2008.

[CKC+08b] 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 *SIGIR*, 2008.

[CL08] Yin-Wen Chang and Chih-Jen Lin. Feature ranking using linear svm. In *Proc. of WCCI Causation and Prediction Challenge*, pages 53–64, 2008.

[CM83] Thomas F Coleman and Jorge J Moré. Estimation of sparse jacobian matrices and graph coloring blems. *SIAM journal on Numerical Analysis*, 20(1):187–209, 1983.

[CMS09] Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice.* Addison-Wesley Publishing Company, USA, 1st edition, 2009.

[CMZG09a] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 621–630, 2009.

[CMZG09b] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 621–630, New York, NY, USA, 2009. ACM.

[CN05] Edgar Chávez and Gonzalo Navarro. A compact space decomposition for effective metric indexing. *Pattern Recognition Letters*, 26(9):1363–1376, 2005.

[CNBM+12] Masashi Crete-Nishihata, Ronald M. Baecker, Michael Massimi, Deborah Ptak, Rachelle Campigotto, Liam D. Kaufman, Adam M. Brickman, Gary R. Turner, Joshua R. Steinerman, and Sandra E. Black.

Reconstructing the Past: Personal Memory Technologies Are Not Just Personal and Not Just for Memory. *HumanComputer Interaction*, 27(1-2):92–123, 2012.

[CNPS11] Gabriele Capannini, Franco Maria Nardini, Raffaele Perego, and Fabrizio Silvestri. Efficient diversification of web search results. *PVLDB*, 4(7):451–459, 2011.

[COA14] Sergiu Chelaru, Claudia Orellana-Rodriguez, and Ismail Sengor Altingovde. How useful is social feedback for learning to rank youtube videos? *World Wide Web*, 17(5):997–1025, 2014.

[CORA] Sergiu Viorel Chelaru, Claudia Orellana-Rodriguez, and Ismail Sengor Altingovde. How useful is social feedback for learning to rank youtube videos? *WWW Journal*, pages 1–29. In press, doi:10.1007/s11280-013-0258-9.

[CORA14] Sergiu Chelaru, Claudia Orellana-Rodriguez, and Ismail Sengor Altingovde. How useful is social feedback for learning to rank youtube videos? *World Wide Web*, 17(5):997, 2014.

[CR12] Federico Cingano and Alfonso Rosolia. People i know: job search and social networks. *Journal of Labor Economics*, 30(2):291–332, 2012.

[CS06] Edith Cohen and Martin J. Strauss. Maintaining time-decaying stream aggregates. *J. Algorithms*, 59(1):19–36, April 2006.

[CVK$^+$10] Berkant Barla Cambazoglu, Emre Varol, Enver Kayaaslan, Cevdet Aykanat, and Ricardo A. Baeza-Yates. Query forwarding in geographically distributed search engines. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 90–97, 2010.

[DBC13] Van Dang, Michael Bendersky, and W. Bruce Croft. Two-stage learning to rank for information retrieval. In Pavel Serdyukov, Pavel Braslavski, Sergei O. Kuznetsov, Jaap Kamps, Stefan Rüger, Eugene Agichtein, Ilya Segalovich, and Emine Yilmaz, editors, *Advances in Information Retrieval*, pages 423–434, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[DC10] Van Dang and W. Bruce Croft. Feature selection for document ranking using best first search and coordinate ascent. In *Proc. of SIGIR'10 Workshop on Feature Generation and Selection for Information Retrieval*, 2010.

[DC12] Van Dang and W. Bruce Croft. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74, 2012.

[DC13] Van Dang and W. Bruce Croft. Term level search result diversification. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 603–612, 2013.

[DCC$^+$03] Susan Dumais, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. Stuff i've seen: a system for personal information retrieval and re-use. SIGIR '03, pages 72–79, 2003.

[Dea09] Jeffrey Dean. Challenges in building large-scale information retrieval systems: invited talk. In *Proceedings of the 2nd International Conference on Web Search and Web Data Mining*, page 1, 2009.

[DJQ$^+$10] Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. An empirical study on learning to rank of tweets. In *Proc. of COLING '10*, pages 295–303, 2010.

[DP09] Marina Drosou and Evaggelia Pitoura. Diversity over continuous data. *IEEE Data Eng. Bull.*, 32(4):49–56, 2009.

[EGV$^+$13] Nicole B. Ellison, Rebecca Gray, Jessica Vitak, Cliff Lampe, and Andrew T. Fiore. Calling all facebook friends: Exploring requests for help on facebook. In *Proc. of ICWSM '13*, 2013.

[ES06] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC '06)*, pages 417–422, 2006.

[ESL11] Nicole B. Ellison, Charles Steinfield, and Cliff Lampe. Connection strategies: Social capital implications of facebook-enabled communication practices. *New Media & Society*, 13(6):873–892, 2011.

[FE08] Adrienne Felt and David Evans. Privacy protection for social networking platforms. In *Proc. of Web 2.0 Security and Privacy*, 2008.

[FKLT10] Lujun Fang, Heedo Kim, Kristen LeFevre, and Aaron Tami. A privacy recommendation wizard for users of social networking sites. In *Proc. of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 630–632, 2010.

[FL10] Lujun Fang and Kristen LeFevre. Privacy wizards for social networking sites. In *Proc. of WWW '10*, pages 351–360, 2010.

[FLSV11] Moran Feldman, Ronny Lempel, Oren Somekh, and Kolman Vornovitsky. On the impact of random index-partitioning on index compression. *CoRR*, abs/1107.5661, 2011.

[GA05] Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. In *Proc. of the ACM Workshop on Privacy in the Electronic Society*, pages 71–80, 2005.

[GAC$^+$13] Ido Guy, Uri Avraham, David Carmel, Sigalit Ur, Michal Jacovi, and Inbal Ronen. Mining expertise and interests from social media. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 515–526, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[GAS$^+$09] Gunnar Aastrand Grimnes, Benjamin Adrian, Sven Schwarz, Heiko Maus, Kinga Schumacher, and Leo Sauermann. Semantic desktop for the end-user (semantic desktop für anwender). *i-com*, 8(3):25–32, 2009.

[GCSMO11] Veronica Gil-Costa, Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Sparse spatial selection for novelty-based search result diversification. In *Proceedings of the 18th International Symposium on String Processing and Information Retrieval*, pages 344–355, 2011.

[GCSMO13] Veronica Gil-Costa, Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Modelling efficient novelty-based search result diversification in metric spaces. *J. Discrete Algorithms*, 18:75–88, 2013.

[GKDD15] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proc. of CHI '15*, pages 1631–1640, 2015.

[GLNP16] Andrea Gigli, Claudio Lucchese, Franco Maria Nardini, and Raffaele Perego. Fast feature selection for learning to rank. In *Proc. of ICTIR '16*, pages 167–170, 2016.

[GLQ$^+$08] Xiubo Geng, Tie-Yan Liu, Tao Qin, Andrew Arnold, Hang Li, and Heung-Yeung Shum. Query dependent ranking using k-nearest neighbor. In *Proc. of SIGIR'08*, pages 115–122, 2008.

[GLQL07] Xiubo Geng, Tie-Yan Liu, Tao Qin, and Hang Li. Feature selection for ranking. In *Proc. of SIGIR'07*, pages 407–414, 2007.

[GMS13] Kambiz Ghazinour, Stan Matwin, and Marina Sokolova. Monitoring and recommending privacy settings in social networks. In *Proc. of the Joint EDBT/ICDT Workshops*, pages 164–168, 2013.

[GS09] Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *Proceedings of the 18th Int'l Conf. on World Wide Web*, pages 381–390, 2009.

[GSB$^+$12] Saptarshi Ghosh, Naveen Sharma, Fabricio Benevenuto, Niloy Ganguly, and Krishna Gummadi. Cognos: Crowdsourcing search for topic experts in microblogs. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 575–590, New York, NY, USA, 2012. ACM.

[HCC11] Lichan Hong, Gregorio Convertino, and Ed H Chi. Language matters in twitter: A large scale study. In *ICWSM*, 2011.

[HGO00] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. *Advances in Large Margin Classifiers*, pages 115–132, 2000.

[HMdR11] Jiyin He, Edgar Meij, and Maarten de Rijke. Result diversification based on query-specific cluster ranking. *JASIST*, 62(3):550–571, 2011.

[Hoc97] Dorit S. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Co., Boston, MA, USA, 1997.

[HSV95] K.U. Hoffgen, H.U. Simon, and K.S. Vanhorn. Robust trainability of single neurons. *Journal of Computer and System Sciences*, 50(1):114 – 125, 1995.

[HW79] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[JK02a] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[JK02b] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[JL95] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proc. of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 338–345. Morgan Kaufmann, 1995.

[Joa02a] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, New York, NY, USA, 2002. ACM.

[Joa02b] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proc. of KDD'02*, pages 133–142, 2002.

[Joa06] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 217–226, New York, NY, USA, 2006. ACM.

[Joh10] B Johnson. Privacy no longer a social norm, says facebook founder. http://www.theguardian.com/technology/2010/jan/11/facebook-privacy/, 2010.

[Joi08] Adam N. Joinson. Looking at, looking up or keeping up with people?: Motives and use of facebook. In *Proc. of CHI '08*, 2008.

[Jon08] William Jones. *Keeping Found Things Found: The Study and Practice of Personal Information Management*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

[Jon10] William Jones. No knowledge but through information. *First Monday*, 15(9), 2010.

[KB06] Maryam Kamvar and Shumeet Baluja. A large scale study of wireless search behavior: Google mobile search. In *Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI*, pages 701–709, 2006.

[KC10] Anagha Kulkarni and Jamie Callan. Document allocation policies for selective searching of distributed indexes. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management*, pages 449–458, 2010.

[KEN38] M. G. KENDALL. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[KHF$^+$09] Sonja Knoll, Aaron Hoff, Danyel Fisher, Susan Dumais, and Ed Cutrell. Viewing personal data over time. In *Proceedings of CHI'2009 Workshop on Interacting with Temporal Data*, 2009.

[KNH$^+$13] Ricardo Kawase, Bernardo Pereira Nunes, Eelco Herder, Wolfgang Nejdl, and Marco Antonio Casanova. Who wants to get fired? In *In Proc. of WebSci '13*, pages 191–194, 2013.

[KNS13] Nattiya Kanhabua, Claudia Niederée, and Wolf Siberski. Towards concise preservation by managed forgetting: Research issues and case study. In *Proceedings of the 10th International Conference on Preservation of Digital Objects*, iPres '13, 2013.

[KNT10] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In Philip S. Yu, Jiawei Han, and Christos Faloutsos, editors, *Link Mining: Models, Algorithms, and Applications*, pages 337–357. Springer New York, 2010.

[KS10] David S. Kirk and Abigail Sellen. On human remains: Values and practice in the home archiving of cherished objects. *ACM Trans. Comput.-Hum. Interact.*, 17(3):10:1–10:43, July 2010.

[KW11] Vaiva Kalnikaitė and Steve Whittaker. A saunter down memory lane: Digital reflection on personal mementos. *Int. J. Hum.-Comput. Stud.*, 69(5):298–310, 2011.

[Lan07] Patricia G. Lange. Publicly private and privately public: Social networking on youtube. *Journal of Computer-Mediated Communication*, 13(1):361–380, 2007.

[LES08] Cliff Lampe, Nicole B. Ellison, and Charles Steinfield. Changes in use and perception of facebook. In *Proc. of CSCW '08*, 2008.

[LGKM11] Yabing Liu, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. Analyzing facebook privacy settings: user expectations vs. reality. In *Proc. of the 11th ACM SIGCOMM Conference on Internet Measurement*, (IMC '11), pages 61–70, 2011.

[Liu09] Tie-Yan Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, March 2009.

[LLS14] Bin Liu, Jialiu Lin, and Norman M. Sadeh. Reconciling mobile app privacy and usability on smartphones: could user privacy profiles help? In *Proc. of WWW '14*, pages 201–212, 2014.

[LLSH14] Jialiu Lin, Bin Liu, Norman M. Sadeh, and Jason I. Hong. Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings. In *Proc. of the 10th Symposium on Usable Privacy and Security, (SOUPS '14)*, pages 199–212, 2014.

[LRdR14] Shangsong Liang, Zhaochun Ren, and Maarten de Rijke. Fusion helps diversification. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 303–312, 2014.

[LZS$^+$13] Jing Liu, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen Hon. What's in a name?: An unsupervised approach to link users across communities. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 495–504, New York, NY, USA, 2013. ACM.

[Mad12] Mary Madden. Privacy management on social media sites. Technical report, Pew Internet and American Life Project, 2012.

[Man13] A. Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29:229–235, 10/2013 2013.

[Mar52] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[MHO08] Craig Macdonald, David Hannah, and Iadh Ounis. High quality expertise evidence for expert search. In *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval*, ECIR'08, pages 283–295, Berlin, Heidelberg, 2008. Springer-Verlag.

[Mit97] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[MJB12] Michelle Madejski, Maritza Johnson, and Steven M Bellovin. A study of privacy settings errors in an online social network. In *Proc. of PerCom '12 Workshops*, pages 340–345, 2012.

[MM31] John A. McGeoch and William T. McDonald. Meaningful relation and retroactive inhibition. *American Journal of Psychology*, 43(4):579–588, 1931.

[MRS08] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[MS99] C.D. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[MS09] V. Mayer-Schönberger. *Delete - The Virtue of Forgetting in the Digital Age*. Morgan Kaufmann Publishers, 2009.

[MSN11] Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. Incremental diversification for very large sets: a streaming-based approach. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 585–594, 2011.

[MSO12] Craig Macdonald, Rodrygo L. T. Santos, and Iadh Ounis. On the usefulness of query features for learning to rank. In *Proc. of CIKM '12*, pages 2559–2562, 2012.

[MW09] Winter A. Mason and Duncan J. Watts. Financial incentives and the "performance of crowds". *SIGKDD Explorations*, 11(2):100–108, 2009.

[NA14] Kaweh Djafari Naini and Ismail Sengor Altingovde. Exploiting result diversification methods for feature selection in learning to rank. In *Proc. of ECIR'14*, pages 455–461, 2014.

[NAK$^+$15] Kaweh Djafari Naini, Ismail Sengor Altingovde, Ricardo Kawase, Eelco Herder, and Claudia Niederée. Analyzing and predicting privacy settings in the social web. In *User Modeling, Adaptation and Personalization - 23rd International Conference, UMAP 2015, Dublin, Ireland, June 29 - July 3, 2015. Proceedings*, pages 104–117, 2015.

[NAS16] Kaweh Djafari Naini, Ismail Sengor Altingovde, and Wolf Siberski. Scalable and efficient web search result diversification. *ACM Transactions on the Web*, 10(3):15:1–15:30, August 2016.

[NKK$^+$18] Kaweh Djafari Naini, Ricardo Kawase, Nattiya Kanhabua, Claudia Niederée, and Ismail Sengor Altingovde. Those were the days: learning to rank social media posts for reminiscence. *Information Retrieval Journal*, Aug 2018.

[NKKN14] Kaweh Djafari Naini, Ricardo Kawase, Nattiya Kanhabua, and Claudia Niederée. Characterizing high-impact features for content retention in social web applications. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume*, pages 559–560, 2014.

[NKTN18] Claudia Niederée, Nattiya Kanhabua, Tuan Tran, and Kaweh Djafari Naini. *Preservation Value and Managed Forgetting*, pages 101–129. Springer International Publishing, Cham, 2018.

[OA14] Ahmet Murat Ozdemiray and Ismail Sengor Altingovde. Query performance prediction for aspect weighting in search result diversification. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1871–1874, 2014.

[OA15] Ahmet Murat Ozdemiray and Ismail Sengor Altingovde. Explicit search result diversification using score and rank aggregation methods. *JASIST*, 66(6):1212–1228, 2015.

[OAC$^+$12] Rifat Ozcan, Ismail Sengör Altingövde, Berkant Barla Cambazoglu, Flavio Paiva Junqueira, and Özgür Ulusoy. A five-level static cache architecture for web search engines. *Inf. Process. Manage.*, 48(5):828–840, 2012.

[PCT06] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *Proceedings of the 1st International Conference on Scalable Information Systems*, 2006.

[PL08] Qiang Wu Ping Li, Chris J.C. Burges. Learning to rank using classification and gradient boosting. MIT Press, Cambridge, MA, January 2008.

[PLD05] Hanchuan Peng, Fuhui Long, and Chris H. Q. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1226–1238, 2005.

[PSL06] Diego Puppin, Fabrizio Silvestri, and Domenico Laforenza. Query-driven document partitioning and collection selection. In *Proceedings of the 1st International Conference on Scalable Information Systems*, page 34, 2006.

[PVK$^+$04] Themistoklis Palpanas, Michail Vlachos, Eamonn Keogh, Dimitrios Gunopulos, and Wagner Truppel. Online amnesic approximation of streaming time series. In *Proceedings of ICDE '04*, 2004.

[RBS10] Davood Rafiei, Krishna Bharat, and Anand Shukla. Diversifying web search results. In *Proc. of WWW'10*, pages 781–790, 2010.

[RD06] Filip Radlinski and Susan T. Dumais. Improving personalized web search using result diversification. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 691–692, 2006.

[Rob77] Stephen E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33:294–304, 1977.

[Ros11] Chris Rose. The security implications of ubiquitous social media. *International Journal of Management and Information Systems*, 15(1), 2011.

[RWJ$^+$95] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[San13] Rodrygo L. T. Santos. *Explicit web search result diversification.* PhD thesis, University of Glasgow, UK, 2013.

[SCAC13] Rodrygo L. T. Santos, Pablo Castells, Ismail Sengor Altingovde, and Fazli Can. Diversity and novelty in information retrieval. In *Proc. of SIGIR'13*, page 1130, 2013.

[SCNSP10] Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. How useful are your comments?: Analyzing and predicting youtube comments and comment ratings. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 891–900, New York, NY, USA, 2010. ACM.

[SDE$^+$06] Leo Sauermann, Andreas Dengel, Ludger Van Elst, Andreas Lauer, and Maus Sven Schwarz. Personalization in the epos project. In *In Proc. of ESWC 2006*, pages 42–52, 2006.

[Shu10] Nakatani Shuyo. Language detection library for java, 2010.

[SLW11] B. Sparrow, J. Liu, and D. M. Wegner. Google effects on memory: Cognitive consequences of having information at our fingertips. *Science*, 333:776–778, 2011.

[SMO10a] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web*, pages 881–890, 2010.

[SMO10b] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Selectively diversifying web search results. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management*, pages 1179–1188, 2010.

[SMO11] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Intent-aware search result diversification. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 595–604, 2011.

[SMO15] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Search result diversification. *Found. Trends Inf. Retr.*, 9(1):1–90, March 2015.

[SO13] Tasos Spiliotopoulos and Ian Oakley. Understanding motivations for facebook use: usage metrics, network structure, and privacy. In *Proc. of CHI '13*, 2013.

[SWS05] Cees Snoek, Marcel Worring, and Arnold W. M. Smeulders. Early versus late fusion in semantic video analysis. In *Proc. of the 13th ACM Int'l Conference on Multimedia*, pages 399–402, 2005.

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.

[Tar72] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[TSH10] Eran Toch, Norman M. Sadeh, and Jason I. Hong. Generating default privacy policies for online social networks. In *Proc. of CHI'10 (Extended Abstracts Volume)*, pages 4243–4248, 2010.

[Tul02] Endel Tulving. Episodic memory: From mind to brain. *Annual review of psychology*, 53(1):1–25, 2002.

[TWC12] Eran Toch, Yang Wang, and Lorrie Cranor. Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems. *User Modeling and User-Adapted Interaction*, 22:203–220, 2012.

[Und57] Benton Underwood. Interference and forgetting. *Psychological Review*, 64(1):49–60, 1957.

[VC12] David Vallet and Pablo Castells. Personalized diversification of search results. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 841–850, 2012.

[VCV12] Saul Vargas, Pablo Castells, and David Vallet. Explicit relevance models in intent-oriented information retrieval diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 75–84, 2012.

[VRB$^+$11] Marcos R. Vieira, Humberto Luiz Razente, Maria Camila Nardini Barioni, Marios Hadjieleftheriou, Divesh Srivastava, Caetano Traina Jr., and Vassilis J. Tsotras. On query result diversification. In *Proc. of ICDE'11*, pages 1163–1174, 2011.

[WF05] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[WFH11] I.H. Witten, E. Frank, and M.A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, 2011.

[WH14] Shengli Wu and Chunlan Huang. Search result diversification via data fusion. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 827–830, 2014.

[Wil13] Jill Wiltfong. Majority (71%) of global internet users “share” on social media sites, 2013. Available at http://ipsos-na.com/news-polls/pressrelease.aspx?id=6254.

[WS98] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.

[WZ09a] Jun Wang and Jianhan Zhu. Portfolio theory of information retrieval. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–122, 2009.

[WZ09b] Jun Wang and Jianhan Zhu. Portfolio theory of information retrieval. In *Proc. of SIGIR*, pages 115–122, 2009.

[XKJ14] Jierui Xie, Bart P. Knijnenburg, and Hongxia Jin. Location sharing privacy preference: analysis and personalized recommendation. In *Proc. of the 19th Int'l Conference on Intelligent User Interfaces*, pages 189–198, 2014.

[XL07] Jun Xu and Hang Li. Adarank: A boosting algorithm for information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 391–398, New York, NY, USA, 2007. ACM.

[YHT+16] Dawei Yin, Yuening Hu, Jiliang Tang, Tim Daly Jr., Mianwei Zhou, Hua Ouyang, Jianhui Chen, Changsung Kang, Hongbo Deng, Chikashi Nobata, Jean-Marc Langlois, and Yi Chang. Ranking relevance in yahoo search. In *Proc. of KDD '16*, pages 323–332, 2016.

[YP97] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. pages 412–420. Morgan Kaufmann Publishers, 1997.

[YSK03] Dawit Yimam-Seid and Alfred Kobsa. Expert-finding systems for organizations: Problem and domain analysis and the demoir approach. *Journal of Organizational Computing and Electronic Commerce*, 13(1):1–24, 2003.

[ZA10] Guido Zuccon and Leif Azzopardi. Using the quantum probability ranking principle to rank interdependent documents. In *Proceedings of the 32nd European Conf. on IR Research*, pages 357–369, 2010.

[ZAZW12] Guido Zuccon, Leif Azzopardi, Dell Zhang, and Jun Wang. Top-k retrieval using facility location analysis. In *Proceedings of the 34th European Conference on IR Research*, pages 305–316, 2012.

[ZCL03] ChengXiang Zhai, William W. Cohen, and John D. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–17, 2003.

[ZHLL12] Xin Zhang, Ben He, Tiejian Luo, and Baobin Li. Query-biased learning to rank for real-time twitter search. In *Proc. of CIKM '12*, pages 1915–1919, 2012.

[ZL06] ChengXiang Zhai and John D. Lafferty. A risk minimization framework for information retrieval. *Inf. Process. Manage.*, 42(1):31–55, 2006.

[ZSHD12] Sergej Zerr, Stefan Siersdorfer, Jonathon Hare, and Elena Demidova. Privacy-aware image classification and search. In *Proc. of SIGIR '12*, pages 35–44, 2012.

[ZSN$^+$13] Xuan Zhao, Niloufar Salehi, Sasha Naranjit, Sara Alwaalan, Stephen Voida, and Dan Cosley. The many faces of facebook: Experiencing social media as performance, exhibition, and personal archive. In *Proc. of CHI '13*, 2013.

CURRICULUM VITAE

2019	**Dr. rer. nat. in Computer Science, Leibniz University of Hannover, L3S Research Center, Hannover**
2010 - 2016	**Research Assistant, PhD Student, L3S Research Center, Faculty of Electrical Engineering and Computer Science, Leibniz University of Hannover, Germany**
2012 - 2013	**Graduate Teaching Assistant, Faculty of Electrical Engineering and Computer Science, Leibniz University of Hannover, Germany**
2007 - 2010	**Undergraduate Teaching Assistant, Institute of Production Engineering and Machine Tools, Leibniz University of Hannover, Germany**
2004 - 2010	**Diplom (e.q. M.Sc.) in Mathematics with the field of study Computer Science, Leibniz University of Hannover, Germany**