Approaches to Automatic Text Structuring

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades Dr.-Ing

vorgelegt von
M.Sc. Nicolai Erbs
geboren in Scherzingen (Schweiz)

Tag der Einreichung: 19. Dezember 2014
Tag der Disputation: 02. Februar 2015

Referenten: Prof. Dr. Iryna Gurevych
Prof. Dr. Eneko Agirre
Prof. Dr. Torsten Zesch

Darmstadt 2015
D17

Abstract

Structured text helps readers to better understand the content of documents. In classic newspaper texts or books, some structure already exists. In the Web 2.0, the amount of textual data, especially user-generated data, has increased dramatically. As a result, there exists a large amount of textual data which lacks structure, thus making it more difficult to understand. In this thesis, we will explore techniques for automatic text structuring to help readers to fulfill their information needs. Useful techniques for automatic text structuring are keyphrase identification, table-of-contents generation, and link identification. We improve state of the art results for approaches to text structuring on several benchmark datasets. In addition, we present new representative datasets for users' everyday tasks. We evaluate the quality of text structuring approaches with regard to these scenarios and discover that the quality of approaches highly depends on the dataset on which they are applied.

In the first chapter of this thesis, we establish the theoretical foundations regarding text structuring. We describe our findings from a user survey regarding web usage from which we derive three typical scenarios of Internet users. We then proceed to the three main contributions of this thesis.

We evaluate approaches to keyphrase identification both by extracting and assigning keyphrases for English and German datasets. We find that unsupervised keyphrase extraction yields stable results, but for datasets with predefined keyphrases, additional filtering of keyphrases and assignment approaches yields even higher results. We present a decompounding extension, which further improves results for datasets with shorter texts.

We construct hierarchical table-of-contents of documents for three English datasets and discover that the results for hierarchy identification are sufficient for an automatic system, but for segment title generation, user interaction based on suggestions is required.

We investigate approaches to link identification, including the subtasks of identifying the mention (anchor) of the link and linking the mention to an entity (target). Approaches that make use of the Wikipedia link structure perform best, as long as there is sufficient training data available. For identifying links to sense inventories other than Wikipedia, approaches that do not make use of the link structure outperform the approaches using existing links. We further analyze the effect of senses on computing similarities. In contrast to entity linking, where most entities can be discriminated by their name, we consider cases where multiple entities with the same name exist. We discover that similarity depends on the selected sense inventory.

To foster future evaluation of natural language processing components for text structuring, we present two prototypes of text structuring systems, which integrate techniques for automatic text structuring in a wiki setting and in an e-learning setting with eBooks.

Zusammenfassung

Ein strukturierter Text hilft Lesern den Inhalt eines Dokuments besser zu verstehen. Bei herkömmlichen Textmedien wie Zeitungsartikeln oder Büchern ist bereits eine Struktur vorgegeben. Im Web 2.0 hat sich die Menge an Texten, insbesondere der von Nutzern erstellten, dramatisch erhöht. Ein großer Teil dieser Texte ist daher unstrukturiert und ihr Verständnis dadurch erschwert. In dieser Arbeit werden wir Techniken zur Textstrukturierung untersuchen, um Lesern bei der Erfüllung ihres Informationsbedürfnisses zu helfen.

Nützliche Techniken für die automatische Textstrukturierung sind die Identifikation von Schlüsselphrasen, die Generierung von Inhaltsübersichten und die Identifikation von Verlinkungen. Wir konnten die Resultate für den aktuellen Forschungsstand im Bereich der Ansätze zur Textstrukturierung bei mehreren der üblichen Datensätze verbessern. Darüber hinaus präsentieren wir neue repräsentative Datensätze für häufige Szenarien, in denen Nutzer nach Informationen suchen. Wir evaluieren die Qualität der Ansätze zur Textstrukturierung in Bezug auf diese Szenarien und stellen fest, dass diese stark von dem jeweils gewählten Datensatz abhängt.

Zu Beginn dieser Arbeit, beschäftigen wir uns mit den theoretischen Grundlagen der Textstrukturierung. Wir erläutern unsere Ergebnisse aus einer Nutzerumfrage zu dem Gebrauch des Internets, woraus wir drei typischen Szenarien von Internetnutzern ableiten. Anschließend beschäftigen wir uns in drei Kapiteln mit den zentralen Inhalten dieser Arbeit.

Wir evaluieren Ansätze zur Identifikation von Schlüsselphrasen, sowohl durch Extraktion als auch durch Zuordnung von Schlüsselphrasen für englische und deutsche Datensätze. Wir beobachten, dass nicht überwachte Ansätze zur Identifikation von Schlüsselphrasen stabile Ergebnisse liefern. Bei Datensätzen mit vordefinierten Schlüsselphrasen werden sie jedoch von Ansätzen mit Filterung oder Zuordnung übertroffen. Wir präsentieren eine Erweiterung dieses Ansatzes, bei dem die Komposita getrennt werden. Hierdurch werden die Resultate bei Datensätzen mit kürzeren Texten weiter verbessert.

Wir konstruieren hierarchische Inhaltsverzeichnisse für drei englische Datensätze und stellen fest, dass die Resultate für die Identifikation der Hierarchie für ein automatisches System ausreichend sind. Allerdings ist für die Generierung von Titeln eine Nutzerinteraktion notwendig.

Weiterhin untersuchen wir Ansätze für die Identifikation von Links. Diese müssen zwei Aufgaben erfüllen, zum einen die Identifikation von Erwähnungen (Anker) des Links und zum anderen Verlinkung der Erwähnung zu einer Entität (Ziel). Ansätze, die auf der Linkstruktur von Wikipedia beruhen, liefern die besten Resultate, sofern genügend Trainingsdaten zur Verfügung stehen. Um Links zu anderen Bedeutungsinventaren zu identifizieren, erweisen sich Ansätze, die nicht auf der Linkstruktur basieren, überlegen gegenüber linkbasierten Ansätzen. Weiter analysieren wir den Effekt von Bedeutungen auf die Berechnung von Ähnlichkeiten. Im Gegensatz zu der Verlinkung von Erwähnungen, wo viele Entitäten anhand ihres Namens unterschieden werden können, betrachten wir Fälle, in denen mehrere Entitäten mit identischem Namen existieren. Wir beobachten, dass die Ähnlichkeit von dem gewählten Bedeutungsinventar abhängt.

Um die zukünftige Evaluation von Komponenten der natürlichen Sprachverarbeitung zur Textstrukturierung zu fördern, präsentieren wir zwei Prototypen von Textstrukturierungssystemen. Diese integrieren Techniken zur automatischen Textstrukturierung in einer Wiki-Umgebung bzw. einem E-Learning Szenario mit eBooks.

Acknowledgements

Writing this dissertation would have not been possible without the support of many people. I would like to express my deepest gratitude to Prof. Dr. Iryna Gurevych for encouraging my research, for allowing me to grow as a research scientist, and pushing me to develop the discipline required to reach my full potential. I especially want to thank my co-supervisor Prof. Dr. Torsten Zesch for the endless support and patience. I could not have asked for a better role model. I would also like to thank my co-supervisor Prof. Dr. Eneko Agirre for his valuable guidance, feedback, and advice during my stay in Donostia and beyond.

This work has been supported by the Klaus Tschira Foundation under project No. 00.133.2008, by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the German Federal Ministry of Education and Research (BMBF) within the context of the Software Campus project *open window* under grant No. 01IS12054. I would like to thank all organizations for their generous support. In addition to that, I am grateful to the team of the Software Campus from the EIT ICT Labs, especially Erik Neumann and Maren Lesche, for providing me with many great opportunities. I would also like to thank all industry partners for the great cooperation and Michael Schmidt from IMC for the fruitful discussions.

I would like to thank my current and former colleagues at the Ubiquitous Knowledge Processing for making the last five years a great experience. I became friends with many of you and I thank you for countless discussions and for your support. Many thanks to Lisa Beinborn and Johannes Daxenberger for their gentle feedback in the last few months and many thanks to Richard Eckart de Castilho, Daniel Bär, Petra Stegmann, and Pedro Santos for making time in the office so entertaining.

I am enormously grateful to all my friends who allowed me to step back and start over with a healthy mind. I thank my volleyball team, Sven Kohoutek, and Simon Forster for keeping me in shape. A special thanks to my family. Words cannot express how grateful I am to you. At the end I would like to express my deepest appreciation to Michèle for supporting me, for indulging me, and for always giving me structure.

Contents

Chapter 1

Introduction

Knowledge is of two kinds. We know a subject ourselves,
or we know where we can find information on it.

Samuel Johnson

Information has become one of the most valuable goods in today's society. With the Internet, information can be transmitted within seconds across the world and people can fulfill their information need globally. Important events are usually covered by several news agencies, which allows for reporting on an event from different perspectives. A large proportion of the world's knowledge is available online. Literature is digitalized by Google Books[1] or the Gutenberg project[2] and made freely available. With the advent of the *Web 2.0*, huge amounts of user-generated data became available. Looking for information on a holiday trip to California reveals not only facts about the state and its interesting sites, but also hotel offers, travel reports, and lots of reviews.

The kind of information available is very heterogenous, some of it is of high quality and written by domain experts, while other content is highly-opinionated or contains many grammatical and orthographical errors. Nevertheless, all texts may contain relevant information and it is a tedious task to find documents fulfilling an information need satisfactory. Depending on the scenario, a user might be interested in different types of text. Somebody interested in news events will search on websites of newspapers or agencies to find high-quality content. Researchers are mostly interested in non-opinionated factual information from academic literature, or Internet encyclopedias. Companies are often interested in users' opinions about their products. And finally, employees of companies are probably interested in specific information about the company, which is only to be found in internal documents.

With so many potentially relevant documents, it becomes vital for users to quickly gain an overview of the documents' contents. A structured representation of a document helps users to better understand what the document is about. Some texts are structured in some way, e.g. books are structured in chapters and sometimes include a table-of-contents. Other texts, e.g. reviews, may contain keyphrases, or tags, to support categorizing them. Encyclopedias[3] and other web documents contain links that connect articles, which allow users to quickly explore a large set of articles for a specific topic. However,

[1] http://books.google.com/

[2] https://www.gutenberg.org/

[3] E.g. the collaboratively constructed encyclopedia Wikipedia.[4]

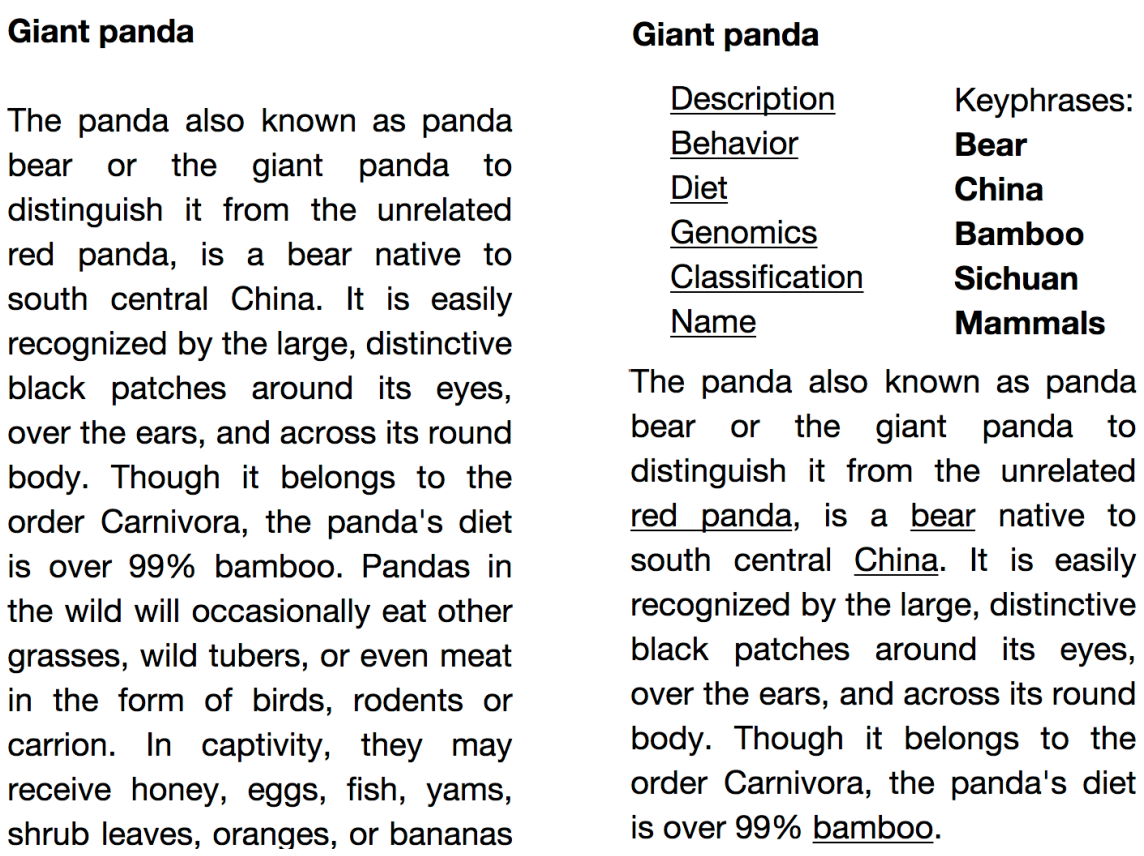


Figure 1.1: The beginning of the altered Wikipedia article about the giant panda. On the left side the text is unstructured and on the right side it is structured with a table-of-contents (on the left), keyphrases (bold), and links (underlined).

not all documents contain this kind of structuring information, thus making it harder to find information in heterogenous documents. Figure 1.1 shows the beginning of the altered Wikipedia article about the giant panda with and without any additional structure. As the article is much longer as shown here, the table-of-contents (left) helps to quickly jump to the most interesting segment, even if it appears much later. The keyphrases (bold) help to quickly get an overview of the content and the links (underlined) enable readers to browse to related articles and understand important related concepts.

Forcing writers to structure every text is infeasible. In corporate scenarios, employees can be required to follow certain guidelines and other employees can be hired to ensure that contributors follow these guidelines. In an open setting, e.g. private websites or blogs, writers cannot be forced to structure their texts. Automatically structuring such texts will allow users to include them into their information search and quicker fulfill their information need.

In this thesis, we analyze keyphrase identification, table-of-contents generation, and link identification as techniques for text structuring. We describe in which scenarios these techniques can be applied in order to structure text. Our main contributions are novel approaches to automatic text structuring. For each of the text structuring techniques, we present approaches and evaluate those on existing and new datasets. The datasets represent different types of documents, which allow for evaluating text structuring approaches in different scenarios.

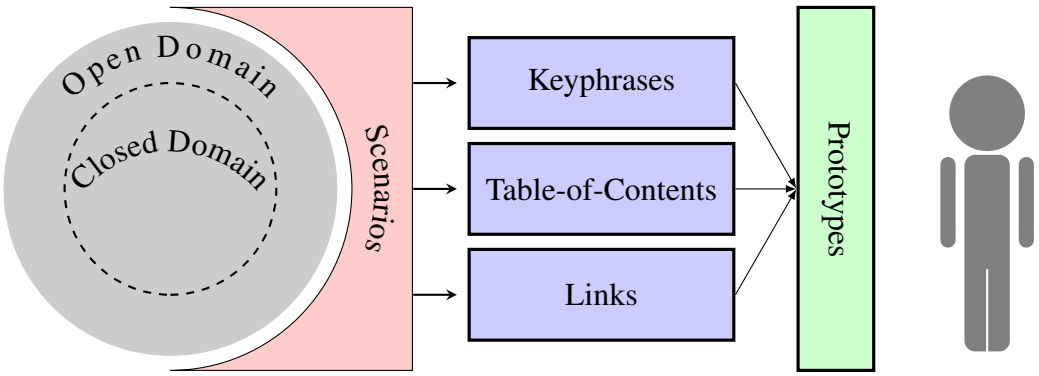


Figure 1.2: Graphical overview of the thesis' contributions. Starting on the left side, the graphic shows the open and closed domain (highlighted light grey) scenarios in which text structuring is beneficial (red), techniques for text structuring which are then shown as the three vertically aligned boxes (blue). Then the prototype systems (green) developed throughout this work visualize the text structure to the user (dark grey) on the right side.

As part of this thesis, we define scenarios, in which approaches to automatic text structuring are beneficial. We define these scenarios based on previous findings and a survey in which we ask Internet users about the ways they use the Internet and the tasks they tackle. These scenarios accompany the evaluation of approaches and allow us to show in which scenarios an approach is best suited for automatic text structuring.

Figure 1.2 shows a graphical structure of the thesis' content. We present different environments (highlighted light grey) and select scenarios in which we test the applicability of techniques for text structuring (highlighted red). We present three techniques for text structuring (highlighted blue), which will in turn be visualized in prototypes (highlighted green) to a user (dark grey).

1.1 Main Contributions

In this thesis we have the following main contributions:

Present scenarios for text structuring: We conduct a survey asking participants about their web usage and based on our findings, we derive three scenarios for search tasks in specific environments.

Develop approaches to keyphrase identification: We present both unsupervised and supervised approaches to keyphrase identification. The unsupervised approaches are based on extracting keyphrases which are present in the document and the supervised approaches are based on assigning keyphrases to a document from a label set. In addition, we present a decompounding extension for keyphrase extraction approaches. For German datasets, we apply decompounding as a preprocessing step and can rely on improved frequency counts.

Develop approaches to table-of-contents generation: We generate a hierarchical table-of-contents by identifying segment titles and putting them in a hierarchical order. We present a supervised approach for the task of hierarchy identification and both supervised and unsupervised approaches for segment title generation.

Develop approaches to link identification: We present an unsupervised alternative for link identification, which contains the subtasks mention identification (finding good anchors) and entity linking (linking to target). Existing supervised approaches rely on the Wikipedia link structure. We further present means to transform a word similarity metric to a sense similarity metric. This is a special case of entity linking, where several entities share the same name.

Analyze applicability of approaches in scenarios: We analyze approaches for text structuring by taking the user scenarios into consideration. One approach might be adequate in one scenario, however lacking required resources in another scenario. We thus analyze which approach is best suitable for a specific scenario. We create representative datasets for these scenarios and evaluate existing and our approaches with these datasets.

Create user-friendly structuring prototypes: We show how to integrate techniques for text structuring in end-user applications. We present two prototypes in which we applied these text structuring applications in educational settings. In Wikulu (Bär et al., 2011a), we integrated components in a wiki setting, which is useful in an education or corporate environment (Buffa, 2006; Ravid et al., 2008; Désilets et al., 2005). With *open window*[5], we integrated a link discovery system in an eBook setting which is designed for Massive Open Online Courses (MOOCs).

1.2 Thesis Organization

In the remainder of this chapter, we give an overview of the organization of this dissertation.

Chapter 2 presents related work which has shown the benefit of additional text structure. We further present typical environments in which a user searches for information and the tasks she/he is faced with. We conduct a user survey to verify previous findings of Internet user behavior. Based on the environments and the tasks, we define three scenarios to represent common searching situations.

Chapter 3 presents keyphrase identification approaches, including extraction and assignment approaches. We analyze the performance of approaches on different datasets: datasets with a predefined set of keyphrases and keyphrases for an open domain. We further evaluate a decompounding extension to improve the results.

Chapter 4 shows how a table-of-contents can be automatically generated to structure a document better. We apply a supervised approach to identify the hierarchy relation of two neighboring segments and apply both unsupervised and supervised approaches for generating titles for the segments.

[5] `https://www.ukp.tu-darmstadt.de/research/current-projects/open-window/` (last accessed: 2014-12-07)

Chapter 5 provides an overview of the process of identifying links and classifies existing approaches. We also evaluate state of the art link identification approaches to Wikipedia articles and then reduce the number of links in the training data to inspect the influence on the approach's results.

Chapter 6 presents two prototypes integrating natural language processing components for text structuring. The first user interface deals with users in a wiki setting, especially corporate wikis. The second user interface integrates a link identification system into an e-learning scenario for eBooks.

Chapter 7 draws conclusions from the preceding chapters and summarizes both the findings of our analysis for text structuring in the defined scenarios and challenges that still remain to be addressed in future work.

Appendix A gives a description of the software packages developed during the course of the thesis.

Chapter 2

Text Structuring

In this chapter, we will present environments and tasks in which text structuring helps users to fulfill their information need. Environments reflect types of textual data a user is dealing with. Based on the environments and tasks we present scenarios to analyze approaches for text structuring. We further introduce techniques for text structuring and their prerequisites in terms of processing input text. We will deal with the following questions to provide the foundation for this thesis:

- Which environments and tasks can be defined to describe Internet users and which scenarios can be defined?
- Which techniques for text structuring do already exist?
- Which preprocessing is needed for automatic text structuring?

A fourth question, which we try to answer in this thesis, is how results for approaches to text structuring depend on the selected scenario. For example, we believe that approaches making use of the domain of the text[1] perform better. Related work has shown that results improve if systems make use of topic or domain knowledge, e.g. the Watson system (Ferrucci et al., 2010) for question answering in Jeopardy,[2] returns significantly better answers after adapting the system to the domain of the questions. Accordingly, we expect that approaches to text structuring largely depend on the task and environment. Thus, we describe common tasks and environments for Internet users.

User survey We conducted a user survey to understand where and why people use the Internet. In addition, we asked participants which kind of techniques for text structuring they would like to use. We recruited volunteers by posting a link to a web-based form[3] on the author's Facebook timeline. The survey contained English questions, however, most of the participants were German. In total, 88 people (47 male and 41 female) took part in the survey. The age of the participants had a range from 19 to 52 years (median of 29 years). Of the 88 volunteers, 59 were employed and 22 were students. The remaining participants were self-employed (3) or did not make any statement (4). Most of the

[1] Assuming that the entire text belongs to a single domain.

[2] `http://www.jeopardy.com/` (last accessed: 2014-12-07)

[3] We used Google Forms `http://www.google.com/forms/about/` (last accessed: 2014-12-07).

participants have an academic background. 15 have a College degree, or similar, as the lowest degree. All of the remaining participants have a higher degree.

2.1 Environments

In this section, we present web environments in which working with texts is important and we describe characteristics for these environments. We compare analysis from related work with findings, which are based on a user survey we conducted.

The Internet has become a huge repository of information, which contains very different types of texts and not only high-quality textual content. Flanagin and Metzger (2001) provide "empirical confirmation that the Internet is a multidimensional communication technology used to fulfill well-understood needs in novel ways". Especially, due to the increasing amount of user-generated data (UGD)[4] and social media in particular, the Internet represents an important source for information. Organizations make use of these social media channels. Kaplan and Haenlein (2010) investigate six types of social media channels: collaborative projects, blogs, content communities, social networking sites, virtual game worlds, and virtual social worlds. They present ways in which companies can efficiently make use of these channels. Dinkelacker and Garg (2001) propose to have a *corporate source* for software development in companies to combine the strengths of open-source development with the intellectual property requirements in companies. Treem and Leonardi (2012) analyze the usage of social media in organizations. They reflect which social media technologies (e.g. wikis) can be used in organizations and which implications their usage has on visibility, persistence, editability, as well as association of co-workers, their communication, and their work. Damianos et al. (2007) present a system for social bookmarking in corporate environments for global knowledge management, other works focus on the individual knowledge worker (Richter and Riemer, 2009; Schneckenberg, 2009).

In this thesis, we go beyond the usage of approaches in a single environment. To better assess the scenarios in which people use the Internet, we conducted a user survey.

User survey results In the survey, we asked in which environments participants use the Internet. Figure 2.1 shows the answer distribution. Almost all participants use the Internet at home (87 or 99%) and at work (81 or 92%). Most of the participants are students or employed, which explains the high number of participants using the Internet at work. The number of participants using the Internet while traveling (71 or 81%) or waiting (70 or 80%) is lower. Only five participants selected the option *others*.

Based on the distribution in Figure 2.1, we identified two environments in which almost all participants make use of the Internet: At work and at home. Additionally, participants use the Internet while traveling and waiting.[5] We further differentiate between two

[4] According to the Organisation for Economic Co-operation and Development (Vickery and Wunsch-Vincent, 2007), UGC needs to fulfill three basic requirements in order to be considered as such: first, it needs to be published either on a publicly accessible website or on a social networking site accessible to a selected group of people; second, it needs to show a certain amount of creative effort; and finally, it needs to have been created outside of professional routines and practices.

[5] We did not ask participants about the device they use when accessing the Internet. However, it certainly influences the way the Internet is used. Nylander et al. (2009) present a study with the focus of using cell phones to access the Internet.

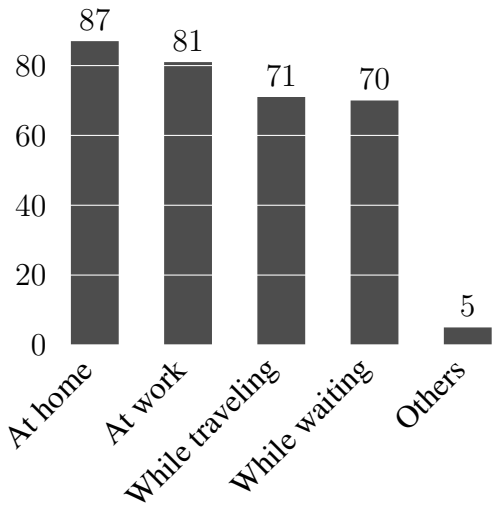


Figure 2.1: Absolute number of answers in our survey for question "Where do you use the Internet?" for four different answer options and a free text field (88 participants and multiple answers allowed).

possible environments of the data: Open domain and closed domain. Information in an open domain environment is not restricted to a single topic or any format, while in closed domain environments, there are requirements regarding topic or style to be met. In this thesis, we will make use of the classification into open domain and closed domain environments. The corporate environment is a special case of the closed domain environment. We will now describe these environments in more detail.

2.1.1 Open domain environment

In general, the Internet is an open domain environment as it covers basically any topic and there are no constrains to website owners regarding the topic or style of a website. Authors of texts are free to express their opinions[6] and they can select any style to do so. This has resulted in many people writing on blogs, using chat programs, or creating whole websites for topics they are interested in.

Recently, the question has been raised whether the Internet is still open and free.[7] The question of a free Internet is out of scope of this thesis as it is rather a political one. However, the question of the Internet being open is valid, as many of the most popular webpages (e.g. Facebook)[8] impose some sort of structure when writing content. Twitter only allows for messages up to 140 characters and Instagram requires a single photo. Still, in terms of textual content, the Internet is an open domain environment as most of the constraints are rather technical ones.

Wikipedia can also be considered an open domain environment. Wikipedia's topics are not restricted, however, there exist policies and guidelines[9] on how to write articles.

[6] With limitations based on the location, e.g. restrictive countries.

[7] A movement of Internet users trying to restrict control of Internet traffic by few companies which created the *Free Internet Act*.

[8] Full list at `http://www.alexa.com/topsites` (last accessed 2014-08-14)

[9] `http://en.wikipedia.org/wiki/Wikipedia:Policies_and_guidelines` (last accessed: 2014-12-07)

It specifies that authors should follow the neutral point of view and it imposes a specific structure, including the commonly used sections *references*, *see also*, and *external links*. Nov (2007) found that the highest motivation for Wikipedia users to contribute is fun[10] and ideology[11], whereas social and career factors seem not to be correlated with the level of contribution.

2.1.2 Closed domain environment

In a closed domain environment, there are typical constraints regarding the topic of content. Examples are fan pages for sports clubs like the Los Angeles Lakers[12] or fictional movie series like the Star Wars fan page[13], which cover only information about a specific topic. People who set up fan pages or contribute to these pages do it because they like to share their interest on these topics.[14]

A collection of many closed domain environments, e.g. a website hosting fan pages about any topic, can be considered an open domain environment.

Corporate environment

The corporate environment is a special case of a closed domain environment.[15] A company creates an environment where employees can freely share information. A very common platform for corporate knowledge sharing are corporate wikis or content management systems. Buffa (2006) shows that using collaborative tools like wikis enables knowledge sharing and creativity. The domain in such an environment is focused on the domain of the company's products. Internal corporate websites and further internal resources are often referred to as *Intranet*.

Contributors in corporate environments sometimes need to be motivated extrinsically in order to contribute in a wiki (Bughin, 2007). Munson (2008) conducted interviews with corporate wiki users and reports that contributors want to manage their reputation in the company, increase the influence of their work, and avoid duplication of efforts. Majchrzak et al. (2006) conducted a survey among users in corporate wikis and suggests the user categories *synthesizers* and *adders*. Synthesizers organize content such as correcting minor mistakes or adding links, adders create content. One difference to other closed domain environments is that in a corporate environment contributors are uncomfortable with editing content they perceive as belonging to others (Munson, 2008). In Wikipedia, on the other hand, conflicting edits of authors are common and thus reverting articles sometimes leads to edit wars (Kittur et al., 2007).

[10] The authors give the example statement: "Writing/editing in Wikipedia is fun."

[11] The authors give the example statement: "I think information should be free."

[12] `http://www.lakersnation.com/` (last accessed: 2014-12-07)

[13] `http://www.theforce.net/` (last accessed: 2014-12-07)

[14] Wilkins (2012) describe the loyalty of professional sport fans.

[15] In case of very large companies, the content increases to a level where it is hard to differentiate between open and closed domain.

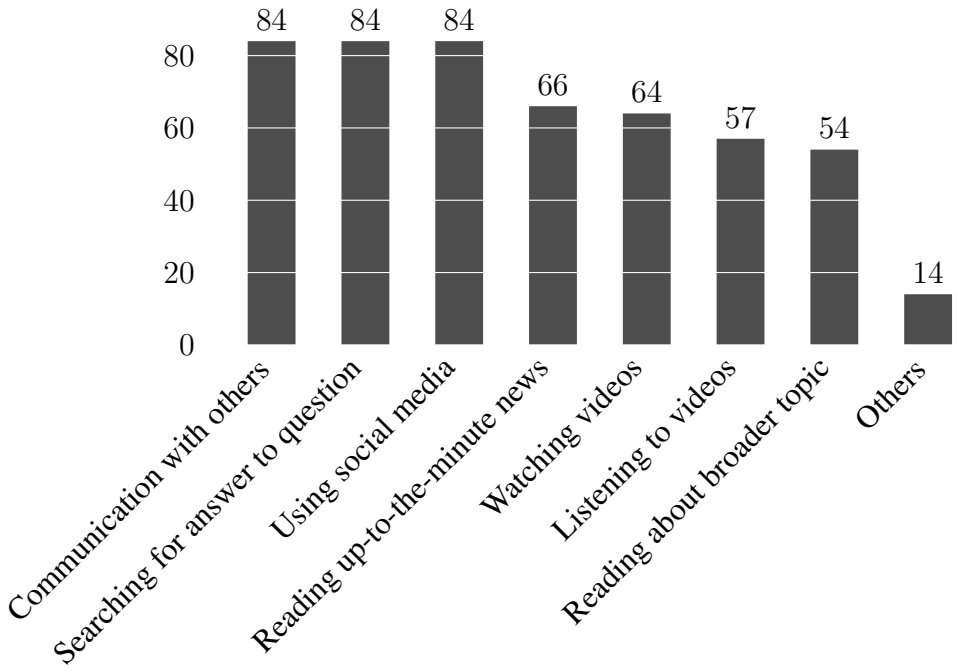


Figure 2.2: Absolute number of answers in our survey for question "If you use the Internet, what do you do?" across seven different answer options and a free text field (88 participants and multiple answers allowed).

2.2 User Tasks

In this thesis, we are interested in the tasks users tackle while using the Internet. In an earlier study, Brandtzaeg et al. (2011) analyze user demographics from 2005 and 2006. They cluster users in five categories: Non-users (42%), sporadic users (18%), instrumental users (18%), entertainment users (10%), and advanced users (12%). These categories partially separate users by the amount of time they spend online. They also separate between the reason for using the Internet as instrumental (goal-oriented activities, such as searching for information about goods or services) and entertainment-related (watching videos, downloading games, or using chat). Singer et al. (2012) follow the categorization with small-scale and large-scale user types oriented towards work, entertainment, and practical information.

When focusing only on heavy web users,[16] Assael (2005) identifies six key web usage categories: web generalists, downloaders, self-improvers, entertainment seekers, stock traders, and socializers. These six categories are partially overlapping with the five categories from Brandtzaeg et al. (2011). Both categorizations share an entertainment-related user type, but they disagree on the broadness of categories. For example, the category *stock traders* is contained in the category *instrumental users* covering work-related user types. In all three categorizations, there exists a category for entertainment-related user types.

User survey results We conducted our own experiment to investigate which tasks Internet users perform. In the user survey presented in Section 2, we asked participants to

[16] According to Assael (2005) these are those using the web for 20 hours a week or more

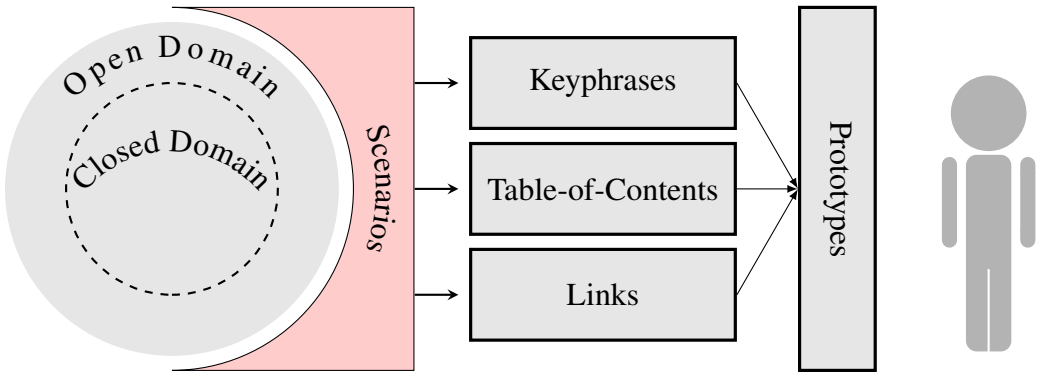


Figure 2.3: We apply techniques for text structuring to scenarios.

select what they do when using the Internet. Figure 2.2 shows the distribution of the participants' answers. Of 88 participants, 84 selected *communication with others*, *searching for answers to a question*, and *using social media*. Our results confirm that many users perform entertainment-related tasks, e.g. *watching videos* (64) and *listening to music* (57). The options *reading about broader topic* (54) and *reading up-to-the-minute news* (66) are—depending on their content—either entertainment-related or instrumental. The option *searching for answer to a question* falls in the instrumental category.

All of the presented studies and our own survey identified entertainment and instrumental tasks as frequently performed activities on the Internet. Based on these findings, we present three scenarios of Internet usage related to textual data, reflecting instrumental and entertainment user types in open domain and closed domain environments as presented in Section 2.1.

2.3 Scenarios

To better analyze approaches to text structuring in several environments and for several user tasks, we define scenarios for Internet usage. Figure 2.3 shows the role of scenarios, which are the connection of text structuring techniques to the textual content in the environments.

Figure 2.4 shows the combination of environments and user tasks with the resulting scenarios. We selected three scenarios for Internet usage related to textual data. The scenarios are influenced by the user types identified in related work and derived from our user survey.

2.3.1 Focused searcher (instrumental user type)

We define this scenario to model a focused Internet user, whose main objective is to get information regarding a specific question or topic. This is a very common scenario in educational settings, where a student needs to answer a question or needs to get informed about a specific topic. A starting point for such a query is very often a search engine. The student then needs to refine the query, or investigate the search results. While investigating the search results from a search engine, there is need for automatic text structuring approaches that allow to decide quickly, (i) whether a document is relevant, (ii) where

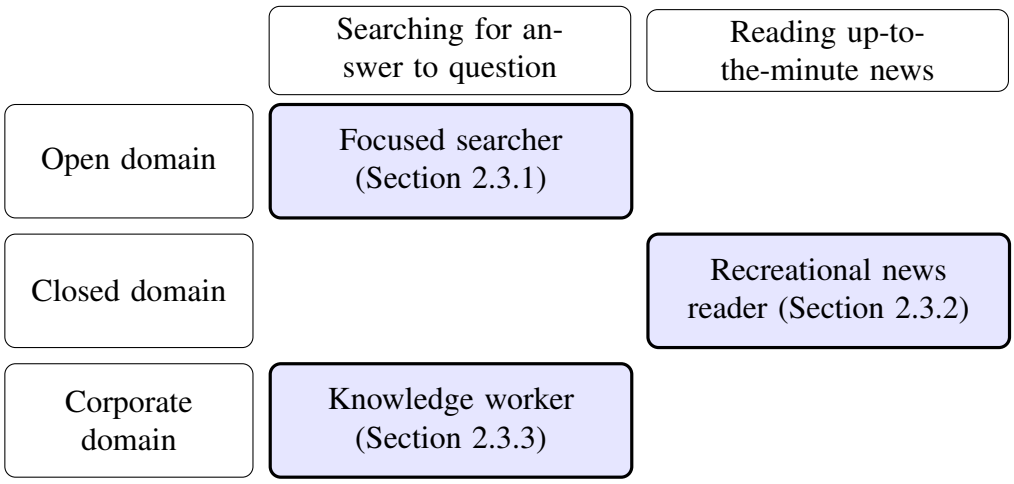


Figure 2.4: Combination of environments and user tasks leads to multiple scenarios.

the relevant information can be found in, and (iii) which other documents may provide further relevant information.

The focused searcher uses open domain environments for starting a search and she/he is likely to end searching in closed domain environments, which are focused on the respective topic. Hölscher and Strube (2000) performed studies regarding the search behavior of experts and new users. Both groups start composing a query in a search engine (81% of less experienced users and 67% of experts), then they rephrase queries, and after examining results rephrase queries again (53% of beginners and 42% of experts). After browsing a document,[17] 72% of all further actions of less-experienced users and 70% of expert's actions correspond to further browsing until the information need is fulfilled. There is a high proportion of continuous browsing because once a website is open, the user continues browsing this website by using the links on a page.

2.3.2 Recreational news reader (entertainment user type in a closed domain environment)

Since searching the web has become easy (Hurtienne and Wandke, 1997), there is an increasing number of people reading news online, while time spent reading newspapers is decreasing (Nie and Erbring, 2000). A recreational news reader has access to the entire news information on the Internet and does not necessarily read news articles in a linear order (from the beginning to the end of the newspaper). On the contrary, the Internet "provides audiences with substantially more control over the news selection process than they enjoyed with the traditional media" (Tewksbury, 2003). However, a recreational news reader tends to stay on one website with many articles, where she/he wants to get informed about world news in general and topic-specific news in more detail. Related work has investigated why users tend to frequently visit certain pages (Li et al., 2006). They use the term *stickiness* to express this habit. They report that stickiness is highly correlated with trust in relation to the web site.

We define this scenario as a user starting her/his information process with a few news

[17] Browsing means using links within the document.

pages, e.g. the BBC[18] for world news, or very specific news pages like MacRumors[19] for news about technology from Apple. The news are from a closed domain because they are topically focused and highly interconnected, which means that a user can easily get 'lost' and learn more about the topic.

2.3.3 Knowledge worker in a company (instrumental user type)

The third scenario is a knowledge worker in a company, most likely in a research and development department. According to Singer et al. (2012), such a user type is a work-oriented Internet user, who is mainly information-oriented. This user consumes information, rather than editing existing text.[20] In contrast to the focused searcher, the knowledge worker is only interested in very detailed information, while the focused searcher needs to get informed about a whole topic.

The information need of a knowledge worker is very often specific to the company, e.g. specification of a company's product, and thus the user will search in the corporate environment. Wikis are a good source of information in corporate environments (Buffa, 2006), because employees can share sensitive information in a corporate wiki without the threat of revealing company secrets to anyone outside of the company.

2.4 Techniques for Text Structuring

Previously, we identified three scenarios for Internet users. For each of the scenarios, a user can apply techniques for text structuring supporting users to deal with their tasks. One of the key contributions of this thesis is to connect text structuring techniques with the scenarios and analyze which approaches work best for each scenario. Thus, we first present existing techniques for text structuring. Techniques for text structuring include—among others—keyphrases, summaries, concept maps, table-of-contents, and links. Text structuring techniques support users in organizing a textual document.[21] In this section, we provide an overview of related work on techniques for text structuring. In particular, we present work showing that structure in a text helps readers to faster and better understand the content of a document.

When looking at a quite uniform collection of documents, people usually rather skim through the collection, instead of reading every single document (Nielsen, 1997). To make information easier to capture, it needs to be highlighted—or simply, be different—in some way. This is "a bias in favor of remembering the unusual", also called the *von Restorff effect* (von Restorff, 1933). Recent studies have verified this effect for highlighting parts of documents. Chi et al. (2007) present an eye-tracking study in which they found that subjects focused on highlighted areas when highlighting cues are present.

There exist a variety of approaches for highlighting data, in particular for textual data.

[18] `http://www.bbc.co.uk/` (last accessed: 2014-12-07)

[19] `http://www.macrumors.com/` (last accessed: 2014-12-07)

[20] Munson (2008) reports that users in corporate environments are uncomfortable with editing content written by others.

[21] In the educational domain, *text structure* also stands for a variety of strategies to comprehend written text (Calfee and Drum, 1986; Stanovich, 2000; Sweet and Snow, 2003).

A continuous zoom[22] is well suited for hierarchical data, e.g. a network plan of a power grid (Bartram et al., 1995). The zoom allows users to see the complete hierarchy and at the same time to focus on one specific area in the hierarchy. For textual data, Fowler and Barker (1974) report that highlighting is suitable because it helps users to better understand the text. Underlining text also helps users to remember facts (without reducing the ability to remember non-underlined text) because it focuses the reader's attention to the most important facts (Hartley et al., 1980; Nist and Hogrebe, 1987).

Other techniques for text structuring include keyphrases, for which Guillory (1998) found that they support foreign language learners to understand videos, vocabulary meanings (McDaniel and Pressley, 1989), and lectures (Barnett, 2003). Concept maps connect a set of concepts according to their relationship in a structured manner (Novak and Cañas, 2008). Concept maps also provide support in learning and research scenarios (Gordon, 1995; Cicognani, 2000). Another technique for better learning results are summaries. Learning performance of a group using good summaries without attending a lecture outperforms a control group attending the lecture (Kiewra et al., 1995). Providing students with a table-of-contents has shown to improve the notes they take in lectures (Kiewra et al., 1995) and has shown to improve test performance (Kiewra et al., 1988).

Structure in a text is not only helpful for users, but is also valuable information for automatic approaches, e.g. text classification. Fürnkranz (1999) exploits links—one technique for text structuring—pointing to a page by extracting the link anchor, the paragraph heading or the whole paragraph text in which the link appears. This is additional context to enrich the text of a page. It allows for classifying documents without any textual content, e.g. pages with images. The PageRank algorithm (Page et al., 1999), which is a fundamental part of Google's search algorithm, is also based on the link structure.

So far, we have presented related work about the usefulness of text structuring techniques. Text structuring is relevant for both automatic approaches and humans. In the following, we present techniques for text structuring that help users to acquire the information contained in a text faster. There are several techniques for text structuring based on extracting parts of the text (keyphrases and headlines), on visualizing information in a different manner (concept maps, summaries, and a table-of-contents), and on enriching the text with additional background information (links and link types). In the following, we will introduce these approaches in more detail, however, in this thesis we will further analyze approaches only for keyphrases, table-of-contents, and links as they are—as we will show—among the most popular text structuring techniques in our user study.

2.4.1 Keyphrases

A very common technique for working with textual documents is highlighting relevant passages in a text with a marker. Fowler and Barker (1974) found that highlighting helps students to better understand the text.[23] Hartley et al. (1980) investigate the effect of underlining and showed that it helps sixth-grade students both for short-term memory and long-term memory. The length of relevant passages varies. We focus on short phrases, so called keyphrases. Keyphrases are extracted from the text to help users by (i) providing a

[22] A continuous zoom (dynamic zoom) increases the size of a particular passage or document, while decreasing the size for more distant text.

[23] They found that actively highlighting passages is better than reading documents, where passages have previously been highlighted. However, both techniques help students.

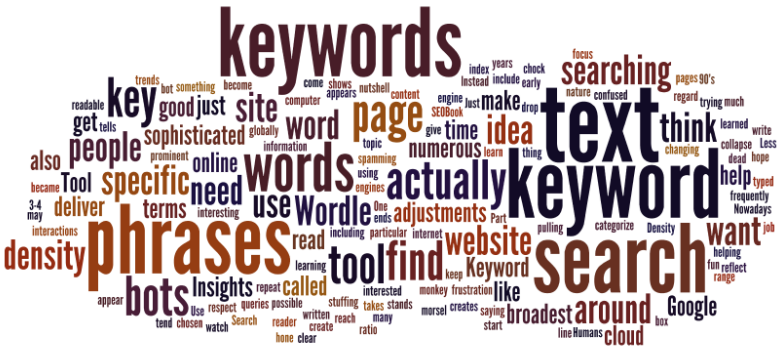


Figure 2.5: A word cloud of keyphrases from text about keyphrases and keywords. Obtained from the web log article `http://beauzartes.com/2011/keywords-keywords-keywords/` with the tool Wordle (`http://www.wordle.net/`) (last accessed: 2014-09-04).

summary of the document (Tucker and Whittaker, 2009), (ii) improving the search process (Song et al., 2006), and (iii) allowing for a dynamic partitioning of topics (Gutwin, 1999).

Figure 2.5 shows a visualization of keyphrases as a word cloud.[24] Such a word cloud is just one possible visual representation of keyphrases. Highlighting or underlining keyphrases are methods used on paper but also in online documents. In this thesis, we will analyze approaches for identifying keyphrases.

2.4.2 Headline generation

The purpose of a headline is twofold: Raising the reader's interest for reading the text, and giving insights about the contents of the text. A headline is either a very short summary of a document, or a short expression to capture the reader's attention. In contrast to a summary, a headline does not necessarily need to follow the grammar of a language. A headline may also be a sentence stub: A concatenation of a few words (e.g. "Obama in Berlin"). Grammatical correctness often is in conflict with the brevity of the headline. Publishers try to bring the reader's attention to the articles with sensation-seeking headlines. Those headlines are not necessarily informative. It is an open question if a headline should be informative or provocative. According to Zajic et al. (2002), the characteristics of headlines can be summarized as follows: (i) Headlines are rather short. They appear to contain between 5 and 15 words. Longer or shorter headlines are unusual. (ii) Headlines reflect the most important facts, which are usually in the beginning of an article. The first few sentences of a text often contain a short description of the whole article. Words from the headline are more likely to appear at the beginning of the text. (iii) Headlines are often built from concatenated words in the text. They form an expression, which should not be split. (iv) Headlines, generated by humans, may be composed of words, which do not appear close to each other in the text (large location gap). Zajic et al. (2002) report that a computer-based system is very bad at generating those headlines. Automatically generated headlines are typically formed by adjacent words and not by words distant to

[24] A word cloud is a visual representation of textual data. The size of the phrase scales with their importance or frequency.

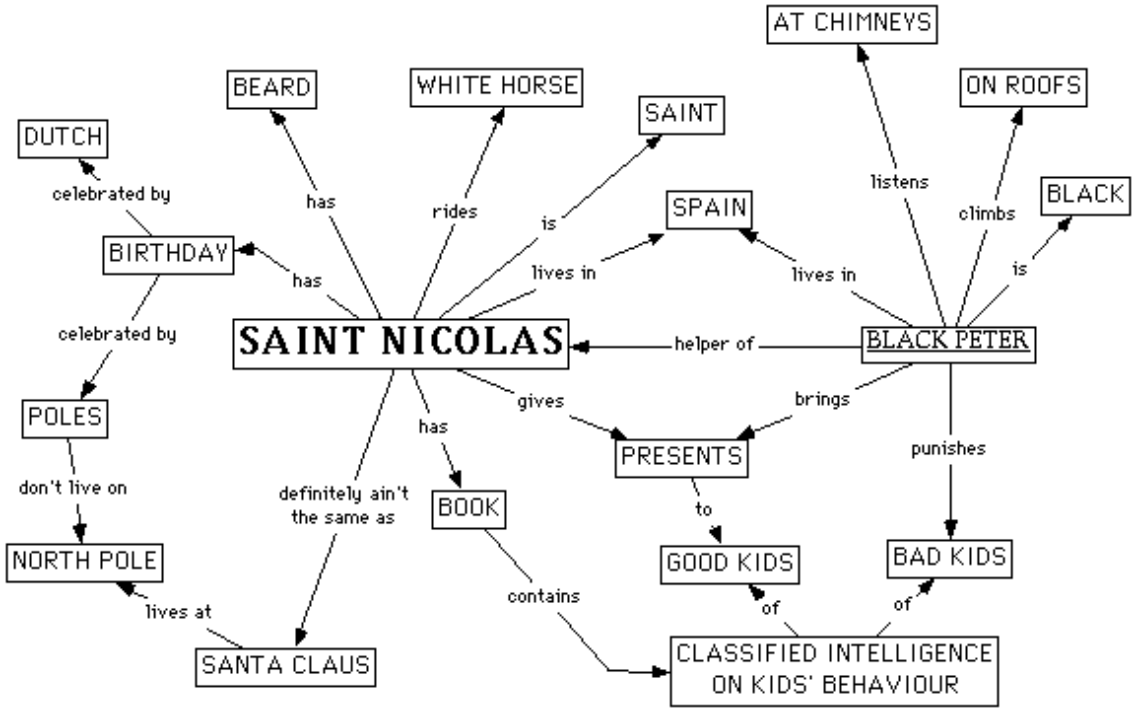


Figure 2.6: Concept map about Saint Nicolas (Lanzing, 1998).

each other in the document.

2.4.3 Concept maps

Concept maps connect a set of concepts according to their relationship in a structured manner. The concepts can be entities or any other phrase with a meaning to the author. A relationship in a concept map defines how two concepts influence or interact with each other, e.g. *is important for*. Novak and Cañas (2008) present a technical report about the foundation of concept maps and include a systematic description about how to use and construct these. Figure 2.6 shows an example of a concept map about Saint Nicolas. It lists figures, places, and things in some way related to *Saint Nicolaus* and shows the relations between them, e.g. Santa Claus *lives at* the north pole. Such a concept map can be perceived as another way to understand a document or a collection of documents.

Such a construct is similar to the Resource Description Framework[25] (RDF), where a statement is constructed by a subject, verb, and object. Furthermore, a concept map allows to combine many RDF-tuples to one connected map. There are numerous tools to create and manage concept maps, promising to help users in managing their knowledge. Examples are the open-source tools iMapping (Haller and Abecker, 2010) and Protégé [26].

The purpose of a concept map is to help users to structure their knowledge. They can also be used to represent information in a text in a structured way, or as Kommers and Lanzing (1997, p. 424) phrase it: "Constructing concept maps stimulates us to externalize, articulate, and pull together information we already know about a subject and understand new information as we learn [...]." The advantages of using concept maps in educational

[25] `http://www.w3.org/TR/PR-rdf-syntax/` (last accessed: 2014-12-07)
[26] `http://protege.stanford.edu` (last accessed: 2014-12-07)

scenarios have been shown in studies with multi-dimensional concept maps (Huang et al., 2012), computer-based concept maps for complex tasks (Tergan, 2004), learning in sport technology (Taşkin et al., 2011), and different coloring techniques (Chiou et al., 2012). Especially with so much information available on the Internet, such a structuring technique helps users in organizing information.

Mind maps (Buzan, 1984) are a similar way to represent the content of a document. They are visualizations of important terms in a hierarchical structure. In contrast to concept maps, they just contain *is-a* relations between their nodes. A hierarchical table-of-contents can be considered as a special representation of a mind map. Like in a mind map, entries in a table-of-contents are ordered in a hierarchy. Additionally, entries in a table-of-contents follow the order of the segments in the document. User generated mind maps can be used to retrieve additional information about the connection between certain topics or important terms. Both, mind maps and table-of-contents should contain no stopwords to use entries as high quality clues for keyphrases and topics.

2.4.4 Segmentation

Segmenting a long text into paragraphs is, according to Purver et al. (2006), the "division of a text or discourse into topically coherent segments". Segmentation is useful in the web when a searcher can be directly guided to the corresponding segment containing the information one is looking for. Otherwise, a user needs to browse manually through the entire document (Salton et al., 1993). Previous work has shown that a segmented text is often easier to process for further natural language processing tasks. Summarization improves on a segmented text, as segments are often created when a new lexical chain starts (Barzilay and Elhadad, 1997).[27] For anaphora detection, segments represent boundaries which are rarely crossed because they could irritate readers (Kozima, 1993).

There are systems for text segmentation based on word reiterations only, such as Hearst's TextTiling (Hearst, 1993) or also Choi's C99 (Choi, 2000). Other algorithms consider word categories from dictionaries such as the approach by Okumura and Honda (1994), or co-occurrence frequencies such as the approach by Ferret (2007).

2.4.5 Table-of-contents

Large documents tend to cover more than one topic, or at least several subtopics. However, a reader is usually not interested in all of those topics to the same extent. The Wikipedia article about *strawberry* (see Figure 2.7) contains many segments and a reader who is interested into nutrition details can directly jump to the 4^{th} segment. An overview with a table-of-contents helps the reader to go directly to the segment about the topic she/he is interested in.

Constructing such a table-of-contents automatically imposes the issues of deciding which hierarchy exists between two segments in the document. For some documents, the hierarchy of segments can be induced using HTML-based features (Pembe and Güngör, 2010), in other cases it can be derived by certain cue phrases such as *back to top*. Feng et al. (2005) train a classifier to detect semantically coherent areas on a page. However,

[27] A lexical chain is a range of text which excels in a high lexical cohesion which means that many of the tokens contained in the chain are lexically related to each other (Morris and Hirst, 1991).

Figure 2.7: The table-of-contents of the English Wikipedia article about *strawberry* from `http://en.wikipedia.org/wiki/Strawberry` (last accessed 2014-09-04).

they make use of the existing HTML markup and return areas of the document instead of identifying hierarchical structures for segments. With hierarchy identification, it is possible to create a hierarchical table-of-contents which enables readers to quickly see the document structure. The idea of a table-of-contents reaches back a long time and was introduced to facilitate the reading of longer books. A table-of-contents gives a brief description of different chapters in a document, optionally with its position in the document.

2.4.6 Link identification

Links in documents help users to quickly get an overview of a broader topic. Users are not restricted to a single document, but can quickly jump to documents giving further information about a mentioned anchor. The anchor represents the source of the link and points to a target. Using link identification, such anchors are automatically identified and linked to the adequate target. Getoor and Diehl (2005) give a survey of different kinds of link identification applications ranging from group identification in social networks to co-reference resolution. In this thesis, we focus on the identification of links from text to other documents. A link connects a mention in the source document to a target document. More than one mention can link to the same target document but a single mention can have at most one target document. A mention usually contains one to four words. Figure 2.8 shows the beginning of the English Wikipedia article about the fruit strawberry. The blue marked phrases lead to further articles inside Wikipedia giving more information about the corresponding phrase.

Strawberry

From Wikipedia, the free encyclopedia

For other species of strawberry, see Fragaria. For other uses, see Strawberry (disambiguation).

The **garden strawberry** (or simply **strawberry** /ˈstrɔːb(ə)ri/; ***Fragaria* × *ananassa***) is a widely grown hybrid species of the genus *Fragaria* (collectively known as the strawberries). It is cultivated worldwide for its fruit. The fruit (which is not a botanical berry, but an aggregate accessory fruit) is widely appreciated for its characteristic aroma, bright red color, juicy texture, and sweetness. It is consumed in large quantities, either fresh or in such prepared foods as preserves, fruit juice, pies, ice creams, milkshakes, and chocolates. Artificial strawberry aroma is also widely used in many industrial food products.

The garden strawberry was first bred in Brittany, France, in the 1750s via a cross of *Fragaria virginiana* from eastern North America and *Fragaria chiloensis*, which was brought from Chile by Amédée-François Frézier in 1714.[1] Cultivars of *Fragaria* × *ananassa* have replaced, in commercial production, the woodland strawberry (*Fragaria vesca*), which was the first strawberry species cultivated in the early 17th century.[2]

Figure 2.8: An excerpt of the English Wikipedia article about *strawberry* from `http://en.wikipedia.org/wiki/Strawberry` (last accessed 2014-09-04).

2.4.7 Link typing

Link typing helps users in understanding what information can be expected before actually opening a page and reading it. This saves time and reduces frustration in case many links lead to less helpful documents for the user. When considering links in text, several types can be distinguished (external links, image links, reference links, ...). Wikipedia pages have special sections listing links following these purposes.

Even links, as they are shown in Figure 2.8 (text phrases linking to a document), may be distinguished in categories. Several previous works undermined the effort to create a taxonomy of link types: Kopak (2000) created a taxonomy differentiating 26 types, which was based on the taxonomies from Trigg (1983), Parunak (1991), and Baron et al. (1996). Kopak's taxonomy contains very specific types which require a manual annotator to have a deep knowledge of the work and the document to be linked, e.g. differentiating between an explanation and a definition. All taxonomies have at least one type of definition link and some differentiate between several types of definitions.

Allan (1996) created yet another taxonomy differentiating three main types of links: *Pattern-matching Links*, *Manual Links*, and *Automatic Links*. This classification is mainly motivated from the programmers' point of view and subject to the capability of the automatic classification approaches. One could consider links created according to simple rules, often referred to as bots, as automatic links.

2.5 Selecting Techniques for Further Analysis

We have presented nine text structuring techniques.[28] Analyzing approaches for all nine techniques in detail would go beyond the scope of this thesis. Thus, we select three

[28] We merge headlines into table-of-contents and we merge Wikipedia and internal links into single categories.

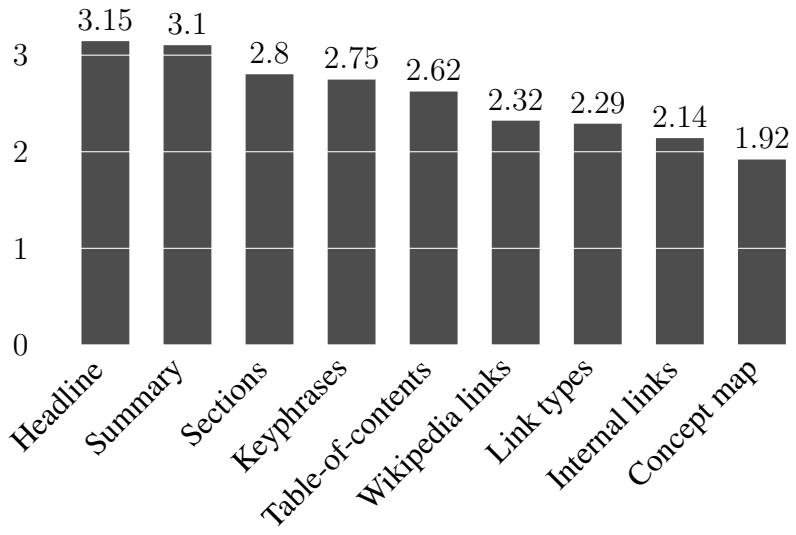


Figure 2.9: The participants' average ratings of different techniques for text structuring on a scale from 0 to 4.

techniques for further analysis. We select them based on previous work (see Section 2.2) and the results from our user survey, which we present in the following.

User survey results In the previously described part of the user survey (see Section 2.1 and Section 2.3), we asked Internet users about environments and tasks when using the Internet. In a separate part of the survey, we firstly asked which techniques for structuring they find useful and secondly, we asked them to rate each technique for text structuring on a scale from 0 to 4. Figure 2.9 shows the ratings of techniques for text structuring.[29] The standard deviation of the ratings is quite high (between .78 and 1.11) which is due to the subjective nature of the question. The use of headlines received the highest rating, and summaries are selected as most useful. Keyphrases and sections are also among the four highest rated and most useful techniques for structuring, followed by table-of-contents and Wikipedia links. Link types, internal links, and a concept map receive a lower rating and are considered less useful by the participants.

According to the findings from the user survey, we disregard concept maps and link types because they were not considered to be helpful by many participants. We will focus on keyphrases as one of the text structuring techniques with a high rating. We will cover headlines and sections as part of the text structuring techniques with table-of-contents. Another focus will be links, primarily to Wikipedia but also, to some extent, to internal sources. We leave summaries to future work.

[29] In a separate question, we asked participants to select useful techniques. Selecting and rating techniques were two distinct questions in the survey, however, there is a strong correlation between rating on a linear scale and assessing techniques for text structuring as useful. The Spearman's rank correlation reaches a value of .91 and Pearson's correlation reaches a value of .90. This shows that users select the highest rated techniques.

2.6 Prerequisites

In order to apply techniques for text structuring, we need to be able to process textual data, thus we will define basic concepts of textual data and describe how to deal with inflection[30] in language. For humans, understanding natural text usually is trivial. Computer algorithms, however, need to perform several steps of analysis in order to determine the meaning of a sentence, e.g. analyzing dependencies and senses. In the first step, automatic processing requires to transform the sequence of characters to a sequence of words. In many languages, a simple splitting at whitespaces is mostly sufficient to identify words. In the following, we describe techniques to identify words, their lemma, and their sense. First, we define the terms *word* and *sense* in more detail.

2.6.1 What is a word?

> *POLONIUS: What do you read, my lord?*
> *HAMLET: Words, words, words.*
> William Shakespeare, *Hamlet*, Act 2, Scene 2, Page 8

A *word* is not consistently defined in related work. As Bauer (1983) points out "The definition of the word has been, for a long time, a major problem for linguistic theory because, however the term word is defined, there are some items in some languages which speakers of those languages call words but which are not covered by the definition." Matthews (1972) defines a word as "what a native speaker thinks a word is". The flexibility of this definition is quite appealing, but it is not clear what happens if two native speakers disagree.

The Merriam-Webster dictionary describes a word as "the entire set of linguistic forms produced by combining a single base with various inflectional elements without change in the part-of-speech elements".[31] This definition reflects lemmas with different suffixes, which are all part of a single word. This corresponds to Peirce's type-token distinction (Peirce, 1906), in which a word type is considered to be the class of its tokens.

The Collins COBUILD Advanced Learner's English Dictionary (Cobuild, 2006) gives the following definition: "[A] word is a single unit of language that can be represented in writing or speech. In English, a word has a space on either side of it when it is written." This definition is more like an instruction how to identify a word in a text based on spaces. This definition is somewhat problematic, as there are many cases when this definition leads to undesired words, e.g. "future—however". Punctuation marks are not considered as spaces, which means that every last word in a sentence will contain a punctuation mark. We have not tackled issues like *don't* which can also be written as two words *do not* or differences in language variants. Depending on the language and the task or context, any definition of a word will be subject to discussion.

Hofland and Johansson (1982) include punctuation marks and hyphens in their definition of a word. This could be driven by the scope of their system, which collects word lists: "A 'word' may contain punctuation marks, as in 2.1 or 3,000. Hyphenated sequences are treated as words [...]." Concerning capitalization they state: "Words which are spelled

[30] Inflection is the modification of a word to express tense, gender, number, or other grammatical categories.

[31] `http://www.merriam-webster.com/dictionary/word` (last accessed: 2014-12-07)

with capitals in all their occurrences in the material are reproduced with capitals in the lists. The others are spelled with lowercase letters." Capitalization is a useful characteristic of English to distinguish between proper and common nouns. However, this does not capture cases when a word is both a proper and a common noun. *Apple* and *Windows* are either the brand or product name if they are capitalized, or the fruit or the part of a house if they are not capitalized.[32]

The example sentence "A rose is a rose is a rose"[33] shows the difficulty of counting words. Some might state it contains eight words, others may say it contains three words. Depending on the definition of a word, both may be correct. A word can be defined as the occurrence or manifestation of a character sequence, or as a unique character sequence. Peirce (1974) called words in the first sense *tokens* and words in the second sense *types*. Lyons (1977, p. 28) claims that the linguist is interested only in types.[34] To differentiate between tokens and words, we will use one definition of word throughout the thesis. We apply the definition of Bauer (1983) for a word:

> A word is all possible representations of the various inflectional categories attached to the base form that is under consideration. For the particular shape that a word has on a particular occasion, the term word form is used. Word forms have a phonological shape or orthographic shape, while a word is a much more abstract unit.

2.6.2 What is a sense?

As opposed to words which are subject to inflections, senses are not inflected. However, different words may have the same sense. Considering the sense of the city of *New York*, the terms *NYC* and *Big Apple* also refer to the same sense.[35] The same words can also have multiple senses, e.g. the word *Washington* has multiple senses, including the first president of the USA, the US state, the capital, or the actor. Words are thus always subject to interpretations. If a word has multiple interpretations, it is called ambiguous, otherwise monosemous. We refer to the different interpretations of words as senses. In cases of real-world objects we also use the term *entity*, e.g. when considering people or cities.

Specifying the sense of some words, especially for verbs and adjectives, is not straightforward. In annotation studies, it has been shown that allowing multiple senses for one word increases inter-annotator agreement. Jurgens (2014) describe that this might be due to the lack of sufficient context available to select a single sense, or due to different interpretations which are possible. In the example sentence "Rooms are classically decorated and warm", the word *warm* may be used in a sense of the comfortable heat level, or in the sense of being colored in such a way to evoke warmth. Very often, there is only a subtle difference between senses, e.g. between the 39 different senses listed for the word *go* in WordNet version 3.1. They differentiate the meaning of go e.g. in the sense of moving and departing. In the example sentence "I go to the bakery to buy some bread", annotators

[32]There are many cases where people are named after common nouns, e.g. the daughter of Kim Kardashian and Kanye West named "North West".

[33]Gertrude Stein in the 1913 poem *Sacred Emily*

[34]This distinction of types and tokens is similar to the distinction of two cars of the same model. Are they actually one car, or are they two cars of the same model?

[35]This may depend on the context, though. One could also talk about a big tasty apple.

might select the sense of *go*, which is related to *move*, or the sense, which is related to *depart*. Depending on the context, both senses can be correct.

In this thesis, we make use of so called *sense inventories* (see Section 5.3.1 for details) to support discrimination of different senses. These sense inventories are lists of senses that can be derived from WordNet (Fellbaum, 1998), Wiktionary (Meyer and Gurevych, 2012b), or GermaNet (Hamp and Feldweg, 1997). Some sense inventories encode words with their part-of-speech along with other linguistic information. In contrast, Wikipedia[36] is more focused on entities (Miller and Gurevych, 2014). It contains less common nouns and more persons, places, and events since it is an encyclopedia.

2.6.3 Dealing with Inflection

> *Morphology is inherently messy.*
>
> Hooper (1979, p. 113)

A very common phenomenon in many languages is inflection. In the example below, the word *goals* is the plural of *goal* and the verb *scored* is the past tense of its base form *score*. Automatic processing needs to map the actual words—the inflected forms—to their base forms. Among others, languages differ to what extent they apply inflection. English is a language, which is only weakly inflected. In contrast to that, German words are inflected more often, some Asian languages use even more inflection. In the German sentence the verb *schoss* (Engl.: *scored*) and the noun *Tore* (Engl.: *goals*) are inflected. The Spanish translation to the given example sentence follows the same grammatical structure. The word *anotó* (Engl.: *scored*) reflects the tense, and the inflected noun *goles* (Engl.: *goals*) represents the number of goals.

English: *The soccer player scored two goals.*

German: *Der Fußballspieler schoss zwei Tore.*

Spanish: *El futbolista anotó dos goles.*

Automatic stemming and lemmatization

We show that stemming and lemmatization are two basic linguistic processing techniques to normalize inflected words, thus enabling better text structuring. The idea of stemming and lemmatization is to map different words to a unified base representation, independent of their inflected forms. With stemming, words are reduced to a stem which is not necessarily a correct word, while a lemma is the base form of a word.

Lovins (1968) claims that automatic stemming is useful for many tasks in Natural Language Processing. He proposes a two-step algorithm using lists of suffixes and transformation rules. The Porter stemmer (Van Rijsbergen et al., 1980; Porter, 1980) is one of the earliest automatic systems for stemming. It is based on hand-crafted rules for reducing words to stems. The Porter stemmer reduces the sentence "The police officers stopped many cars for speeding" to "The polic offic stop mani car for speed". Stemming *many* creates the non-existent word *mani* and the stem of *officers* is *offic*, which is—among others—the stem of *office*.

[36] `en.wikipedia.org/` (last accessed: 2014-12-07)

Lemmas are the base form of word, but many words are ambiguous, e.g. the term *saw*. It can either be the past tense of *see* or the tool. In the example sentence "Yesterday, I saw Jim", the lemma of *saw* is *see* because it is used as a verb. Opposed to stemming, with lemmatization, first, the part-of-speech tag of a word needs to be identified and then the lemma is assigned. This enables dealing with syntactic ambiguity, which is a special case of ambiguity that can be resolved by considering the syntactic structure of a sentence. The term *saw* can either be present tense of the verb *see* or the noun for the tool. Thus, lemmatization includes part-of-speech tagging enabling a system to identify the base form.

Stemming and lemmatization in text processing

Bubenhofer (2009) discusses the usefulness of applying linguistic preprocessing (including lemmatization) to NLP tasks. Using lemma frequencies, instead of token frequencies has the potential to improve performance for several tasks, including searching. Tognini-Bonelli (2001) argues that this means loosing information. She presents the examples *facing* and *faced*, which have both the base form *face*. Their meaning differ depending on the context in which they are used. Tognini-Bonelli (2001) uses the Birmingham corpus and a combination of the Economist and Wall Street Journal corpus. The term *facing* is in most cases used with the physical meaning indicating position and direction (e.g. *facing the table*). The term *faced* is almost exclusively used in a sense indicating the connection with problems and difficulties (e.g. *faced crisis*). Tognini-Bonelli (2001, p. 94) further states:

> "One glance at the collocation profiles [...] dispels any possible illusion that inflected forms are grammatical variations of certain base forms, but broadly share the same meaning of the base form and have a similar behavior."

In her opinion, inflected forms may share the same lemma and have a similar meaning, but reducing them to their lemma would partially remove their meaning. The example of *faced* and *facing* supports this opinion, however, one could also consider that both forms share the same lemma, but have different senses. Thus, the issue of reducing inflected forms to their lemma is the loss of information about their sense. Sinclair (1991, p. 7) tackles this issue by pointing out that form determines the meaning:

> "Soon it was realized that form could actually be a determiner of meaning, and a causal connection was postulated, inviting arguments from form to meaning. Then a conceptual adjustment was made, with the realization that the choice of a meaning, anywhere in a text, must have a profound effect on the surrounding choices. It would be futile to imagine otherwise. There is ultimately no distinction between form and meaning."

Sinclair argues that the form of a word includes valuable information for further processing. It is important to use lemma information as additional information (e.g. for resolving syntactic ambiguity), but it is equally important to keep the form of the word as a valuable source for identifying its meaning. For information retrieval, reducing query words to their lemma is a good starting point, because a relevant document might contain

Language	#Tokens	#Types	TTR	#Lemmas	$\frac{\#\text{Lemmas}}{\#\text{Tokens}}$	#Stems	$\frac{\#\text{stems}}{\#\text{tokens}}$
English	1,771	530	.30	482	.27	481	.27
German	1,777	645	.36	616	.35	574	.32
Spanish	2,086	558	.27	516	.25	485	.23
Turkish	1,430	691	.48	668	.47	577	.40

Table 2.1
Use of inflection in languages in the united declaration of human rights.[37]

the words in a different tense. However, reducing the entire query and all documents to lemmas, impedes the extraction of keyphrases as they might be ungrammatical. Having annotations with additional linguistic information, e.g. as described in Apache UIMA (Ferrucci and Lally, 2004), allows for adding information without removing existing information.

2.6.4 Influence of inflection on corpus statistics

We will now analyze how stemming and lemmatization affect statistics of a corpus across different languages.

Table 2.1 shows inflection statistics in the united declaration of human rights[37] (UDHR). The size of the UDHR is similar across languages. The Spanish edition has the most and the Turkish edition has the fewest tokens. The type-token-ratio (TTR)[38] is lowest for Spanish and highest for Turkish, even though the Spanish dataset contains more tokens. This shows that the variety of different tokens is higher for Turkish, thus showing that Turkish is highly inflected. The type-token-ratio is lower for English than for German with almost the same number number of tokens (1,771 compared to 1,777). A higher type-token-ratio shows that a language contains more inflected words.

The number of different lemmas[39] is lower than the number of types because different types of the same lemma are now counted as one. Again, the ratio of lemmas per tokens is lowest for English and highest for Turkish. The same applies to the number of different stems.[40] The number of stems is lower than the number of lemmas, because rule-based stemming might reduce two words with different senses to the same stem, e.g. the Porter stemmer reduces both *information* and *informed* to the stem *inform*. The difference of number of lemmas and number of stems is smallest for English (only one) and largest for Turkish (approximately 14% less stems than lemmas). A larger difference shows that using lemmatization instead of stemming has a stronger effect on the processed text. In time-critical applications, stemming has the advantage of faster processing, but in less time-critical applications, lemmatization yields better results (Kettunen et al., 2005; Toman et al., 2006). We thus use lemmatization for all further experiments, especially considering the difference of 8% fewer stems in German.

[37] Taken from `http://nltk.googlecode.com/svn/trunk/nltk_data/packages/corpora/udhr.zip` (last accessed: 2014-12-07)

[38] Number of types (set of all words) divided by number of tokens (all words with duplicates).

[39] Identified using the Stanford Core NLP lemmatizer (Manning et al., 2014) from DKPro Core (Eckart de Castilho, 2014)(version 1.5.0) with standard configuration.

[40] Identified using the Snowball stemmer from DKPro Core 1.5.0 with standard configuration.

2.7 Chapter Summary

The main contributions of this chapter can be summarized as follows:

Contribution 2.1: *We presented a classification into open domain and closed domain environments.*
Contribution 2.2: *We conducted a user survey assessing typical tasks of Internet users.*
Contribution 2.3: *We defined scenarios, for which we will analyze the approaches to text structuring in this thesis.*
Contribution 2.4: *We presented techniques for text structuring and selected three for further analysis in this thesis.*
Contribution 2.5: *We defined prerequisites in terms of natural language processing to apply approaches to text structuring.*

In this chapter, we have presented environments and tasks that Internet users tackle. We analyzed related work and conducted an independent user survey and defined scenarios to further analyze strengths and shortcomings of text structuring approaches in each scenario. We defined three scenarios: (i) a focused searcher who is interested in answering question, or finding information about a specific topic, (ii) a recreational news reader looking for information about a broader topic, and (iii) a knowledge worker in a company looking for very specific information.

We described techniques for text structuring and presented work which has shown to support users in several tasks. Keyphrases and table-of-contents (which requires segments and headlines) have shown to help users in better grasping the content of a document. Link identification and typing help users to inspect further relevant documents. Concept maps provide another representation of the information in a document. We selected keyphrase identification, table-of-contents generation, and link identification for further analysis in this thesis.

In addition, we describe prerequisites for preprocessing text to enable approaches to automatic text structuring. Approaches to automatic text structuring may require annotations for words, lemmas, or senses for best results. We have shown that the effect of preprocessing differs depending on the processed language.

Chapter 3

Text Structuring through Keyphrases

> *The assumption of a correlation between the frequency of a linguistic feature and its significance in a text or corpus is one of the basic principles of corpus linguistics.*
>
> Sinclair (1991)

In this chapter, we deal with text structuring through keyphrases as a way to provide a reader with insights into the document's content. Figure 3.1 shows keyphrases as one of the techniques for text structuring.

3.1 Introduction

Sinclair mentions one of the principles that will guide us through this chapter: The frequency of a phrase (or any other selected linguistic feature) correlates with its significance. In this context, significant phrases are those that reflect most of the document's content, or in other words, are keys to the document. In the remainder of this chapter we will thus use the term *keyphrase*. It covers words, or a sequence of words, which are also refereed to as *keywords* (Toolan, 2004), *key words* (Scott, 1996), and *index terms* (Erbs et al., 2013a) that are keyphrases in the special use case of libraries.

Keyphrases are useful as they provide a summary of the document (Tucker and Whittaker, 2009), which makes it easier to decide whether a document contains relevant information for a reader. Keyphrases improve searching (Song et al., 2006) because of keyphrases of higher ranked documents match the query. They also allow for browsing large document collections by their topics, which are defined by their keyphrases (Gutwin, 1999). Typically, only a small fraction of documents share the same keyphrases which allow for fast searching of the whole collection (Song et al., 2006).

Amazon uses *statistically improbable phrases*[1] to give a glimpse into the contents of a book. Toolan (2004) states that keyphrases "[...] are pointers to the topics and thus the content of a literary text". Automatic identification of keyphrases is not only useful in an end user application, but also for research. Baker (2004, p. 347) points out that keyphrases "direct the researcher to important concepts in a text".

Scott (1996, p. 53) consolidates Sinclair's quote about the correlation of frequency and significance by defining keyphrases as those which frequency is unusually high in

[1] Statistically improbable phrases (SIP) are bigrams, which appear infrequently `http://www.amazon.com/gp/search-inside/sipshelp.html` (last accessed Sep. 30, 2014).

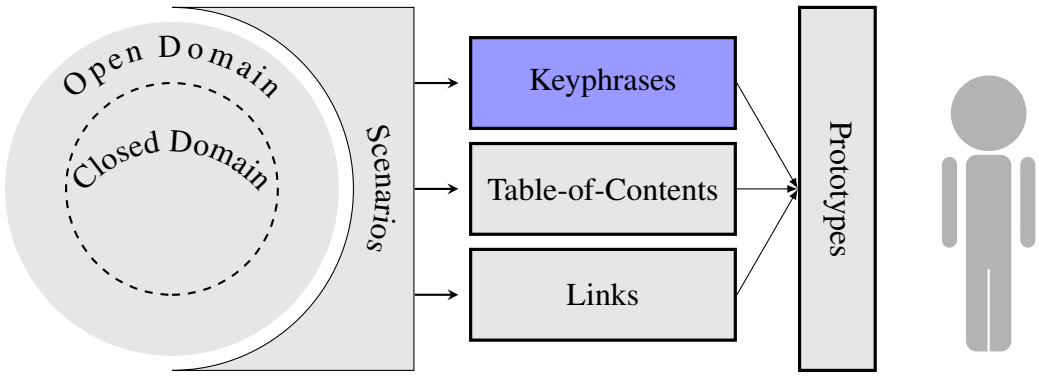


Figure 3.1: Graphical overview of the contents which are covered in this thesis with text structuring through keyphrases highlighted blue.

comparison with some norm. He later specifies this by stating that not the frequency itself is important, but rather an unusual high frequency: A keyphrase "[...] may be defined as a word which occurs with unusual frequency in a given text. This does not mean high frequency but unusual frequency, by comparison with a reference corpus of some kind" (Scott, 1997, p. 236). The choice of a reference corpus is obviously an important criterion, hence the selection of a keyphrase is never absolute. Keyphrases "[...] do not form absolute lexical patterns in a text or corpus, as they depend on the choice of the reference corpus." (Fischer-Starcke, 2010, Chapter 5, p. 65)

The process of identifying keywords used to be a manual process. Ladendorf (1906) created the *Historisches Schlagwörterbuch* (Engl.: *book of historic keywords*), and Lepp (1908) compiled a list of headwords of the reformation period. The lists consist of phrases, which have been popular in the respective time, but have been used less frequently later. In addition to the word lists, they give description and usage examples. For example, Lepp lists the German word *Sophist* (Engl.: *Sophist*) for a person arguing in a hairsplitting manner. Eventually, the process of identifying keywords became an automatic process, making use of computers to count frequencies of phrases. In the following, we will define the task in more detail and present our approaches for automatically identifying keyphrases.

3.2 Task Definition and Characteristics

The identification of keyphrases in documents is mathematically defined as creating a set of phrases (or keyphrase candidates) $p \in P$, for which every phrase has a score $s(p, d)$ typically—but not necessarily—between 0 and 1. The score depends on the phrase p and the document d.[2] The set of phrases will be ranked according to this score and the top-n phrases or the ones above a certain threshold will be considered keyphrases. Figure 3.2 gives an example of a document with keyphrases. The keyphrases on the right side have a score assigned to them, thus leading to a ranked list.

In keyphrase identification, we distinguish between extraction and assignment approaches. For keyphrase extraction, a phrase needs to appear in the document text, while for keyphrase assignment, the phrase does not necessarily need to appear in the document

[2] Approaches may have further parameters, which we will describe for every approach in detail.

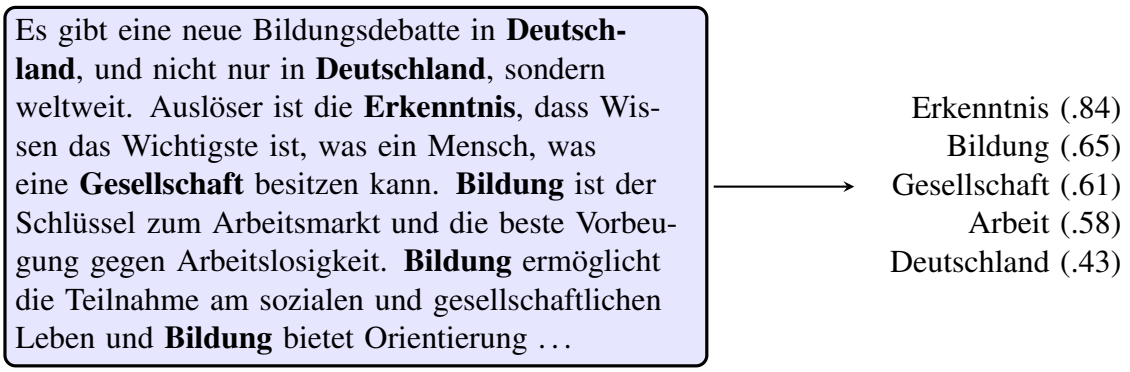


Figure 3.2: A document from the peDOCS dataset with German text (left) and five manually identified keyphrases with scores (right). If an identified keyphrase appears in the document text, it is marked bold. The term *Arbeit* only appears as part of other words in the snippet.

text. In the assignment case, a list of keyphrases must be available. In an extreme case scenario, such a list contains all possible phrases. However, it is obvious that using such a list is unfeasible if not impossible.

The distinction in extraction and assignment approaches has implications on the environments for which these approaches are suitable. In an open domain environment, it is basically impossible to create a keyphrase inventory used for assigning keyphrases. On the contrary, it is possible to create such a keyphrase inventory for a closed domain environment with limited effort. Especially in a corporate environment, it is feasible to construct such a list. In the following section, we further describe both types of approaches and give examples for both types.

3.3 Resources

In this section, we describe datasets for keyphrase identification consisting of textual documents and keyphrases for these documents. The English datasets are commonly used ones and the German datasets have been constructed in the course of this thesis. The English datasets are *Inspec* (Hulth, 2003), *DUC-2001* (Wan and Xiao, 2008a,b), and *SP* (Scientific Papers) (Nguyen and Kan, 2007). We further created three German datasets: *peDOCS*, *MedForum*, and *Pythagoras*.

We will first describe the structure of all datasets in detail and then analyze characteristics of the datasets in a quantitative analysis.

3.3.1 Inspec, DUC-2001, and SP

All three English datasets (Inspec, DUC-2001, and SP) are commonly used datasets for evaluating keyphrase experiments (Mihalcea and Tarau, 2004a; Hasan and Ng, 2010; Zesch, 2009; Hasan and Ng, 2014). This enables us to compare experimental results for a large variety of approaches. These corpora include the document text and a list of keyphrases for every document. They differ by means of which type of text in the document and the method the keyphrases are annotated.

The *Inspec* dataset, introduced by Hulth (2003), consists of 2,000 abstracts of journal

papers from the years 1998 to 2002. They are extracted from the Inspec[3] database, a collection of journals for research literature in physics and engineering. The abstracts were taken from the disciplines "Computers and Control" and "Information Technology". For every document, there exist two sets of keyphrases annotated by professional indexers: (i) controlled terms, and (ii) uncontrolled terms. Controlled terms are keyphrases that are present in an Inspec thesaurus, while uncontrolled terms do not follow this restriction. Thus, the proportion of terms included in the text is lower for controlled terms (18.1%), as opposed to many (76.2%) uncontrolled terms.[4] An example for an abstract (including the title in the first paragraph) is the following:

> Adaptive **state feedback** control for a class of linear systems with unknown bounds of uncertainties
>
> The problem of **robust stabilization** for a class of **linear time-varying systems** with disturbance and **nonlinear uncertainties** is considered. The bounds of the disturbance and uncertainties are assumed to be unknown, being even arbitrary. For such **uncertain dynamical systems**, the adaptive robust **state feedback** controller is obtained. And the resulting **closed-loop systems** are asymptotically stable in theory. Moreover, an adaptive robust **state feedback** control scheme is given. The scheme ensures the **closed-loop systems** exponentially practically stable and can be used in practical engineering. Finally, simulations show that the control scheme is effective.

In the shown example, the uncontrolled keyphrases are marked bold and the controlled keyphrases are underlined. The keyphrases in the set of uncontrolled terms are not influenced by a potentially incomplete or outdated thesaurus. Thus, we use the uncontrolled gold standard[5] for evaluating keyphrase identification approaches.

Over and Yen (2004) originally created the *DUC-2001* dataset for evaluating summarization systems. Later, this dataset was extended with keyphrase annotations (Wan and Xiao, 2008a,b). Two graduate students annotated keyphrases in all documents in the dataset and then resolved conflicts through discussion. The reported Kappa inter-rater agreement of .70 (Wan and Xiao, 2008a) can be considered substantial. No further resources, such as a thesaurus, were used in the annotation process.

The third English dataset we use in this thesis is composed of scientific papers. It is often referred as *NUS Keyphrase Corpus*. We refer to it as *SP* (Scientific Publications) to—uniformly with the other datasets—depict the source of the text. Nguyen and Kan (2007) constructed the SP corpus from 211 documents obtained by searching with Google for PDFs matching the search query "keywords general terms". Only scientific conference papers with a length between 4 and 12 pages were considered. All PDFs were converted into plain text documents and manually annotated by volunteers. Every volunteer annotated three documents, and the union of all annotations for every document compose the final gold standard.[6] We use a random subset of 134 documents from the complete dataset in our experiments (omitting the previously seen development set).

[3] `http://www.theiet.org/resources/inspec/` (last accessed: 2014-12-07)

[4] This is self-evident because there are only few terms (if there are any at all) which are both in the text and in the thesaurus.

[5] "A gold standard dataset or corpus is the one, whose annotation has been checked and corrected. This is typically carried out in order to evaluate automatic annotation systems [...]." (Baker et al., 2006, p. 78)

[6] No statistics regarding inter-annotator agreement were given.

Vocabulary	# Terms	keyphrases covered
Extended thesaurus	8,487	.499
Core thesaurus	974	.263

Table 3.1
Statistics of controlled vocabularies (thesauri) for annotating keyphrases.

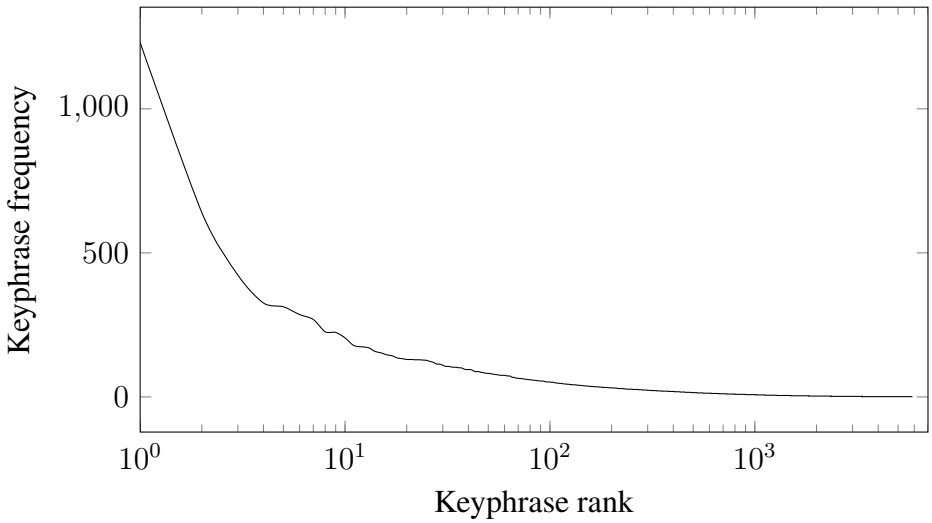


Figure 3.3: The frequency distribution of keyphrases in peDOCS follows a power-law distribution.

3.3.2 peDOCS

peDOCS consists of peer-reviewed articles, dissertations, and books from the educational research domain published by researchers. We first introduced this dataset in Erbs et al. (2013a). We extract all documents from the database dump of *peDOCS*[7] and select all German documents (91% of all documents). Documents span all topics related to education, e.g. historical and general education, pedagogy of media, and environment. They are a valuable resource for teachers looking for teaching material and researchers as a starting point for their research. Hence, the collection is heterogeneous in terms of style, length, and level of detail. Professional indexers assigned keyphrases for every peDOCS document. Due to the size of the corpus, every document was annotated by only one indexer. All indexers follow certain guidelines and apply them to every document in the collection, and the keyphrases should thus be of high quality.

Controlled vocabularies

In addition to the documents of the peDOCS dataset, professional indexers have two thesauri from which they can select keyphrases. Indexers are not restricted to keyphrases from these thesauri; however, the thesauri were constructed by constantly adding frequently assigned keyphrases. This leads to the construction of a *core thesaurus* with 974 terms and an *extended thesaurus* of 8,487 terms. The core thesaurus captures 26.3% and the extended thesaurus captures 49.9% of all identified keyphrases.

[7] `http://www.pedocs.de/` (last accessed: 2014-12-07)

Driven by the high coverage of terms in the set of gold standard keyphrases, we further analyze the frequency of keyphrases in peDOCS. Figure 3.3 shows the frequency distribution of keyphrases. We first rank keyphrases according to their frequency and then plot the resulting distribution. It follows a power-law distribution which proves that few keyphrases are used for many documents and many keyphrases are used only once. Examples of the most frequently used keyphrases are *Deutschland* (Engl.: *Germany*), *Schule* (Engl.: *school*), and *Schüler* (Engl.: *pupils*). These rather general terms do not capture the specific topic of the document, but give readers a coarse-grained classification of the document. A possible use case is to filter a list of keyphrase candidates according to these frequently assigned keyphrases in order to create a clustering of the document collection.

3.3.3 MedForum and Pythagoras

We present two novel keyphrase datasets consisting of German texts. *MedForum* is composed of posts from a medical forum.[8] Users of this forum describe their medical problems, seek for advice, give advice, and discuss medical experiences. We selected only posts with a length of 700-800 characters. The following quotation shows the beginning of a critical post about acupuncture:

> "Ich kann mir irgwie nie so richtig Vorstellen das Akkupunktur wirksam ist! lg Es gibt ein paar enge Indikationen bei denen es eine gewisse Wirksamkeit zu geben scheint. Dabei wurde aber auch schon mehrfach nachgewiesen, daß es völlig egal ist, wohin die Nadeln gestochen werden, also letztendlich, daß dieser mystische Überbau mit Fluss des Chis völliger Humbug ist. [...]"

The post shows that the dataset cannot be considered clean. It contains spelling errors and informal words, such as *Humbug* (Engl.: *nonsense*). To our knowledge, it is the first dataset with keyphrase annotations from user-generated data in German. It allows for a more realistic evaluation of keyphrase identification approaches in the social media environment where text quality usually is lower, e.g. due to wrong grammar and misspelled words (Eisenstein, 2013; Drouin and Davis, 2009; Walther and D'Addario, 2001; Brody and Diakopoulos, 2011; Dresner and Herring, 2010).

Two German annotators with university degrees identified a set of keyphrases for every document and following Nguyen and Kan (2007), the union of both sets are the final gold keyphrases. For training, the annotators first annotated three documents and discussed differences. These documents were later excluded from the final dataset.

The *Pythagoras* dataset contains summaries of lesson transcripts compiled in the Pythagoras project.[9] We selected those documents that were transcriptions of lessons in the German classes (documents from the Swiss German classes were dismissed due to language variations).[10] Two annotators identified keyphrases after a training phase with a discussion of three documents. Both annotators were undergraduate students and native German speakers.[11] As in the MedForum dataset, the gold standard consists of the union of lemmatized keyphrases from both annotators.

[8] http://www.medizin-forum.de/

[9] http://www.dipf.de/en/research/projects/pythagoras

[10] Instead of working on the transcribed lecture, we decided to avoid any issues with analyzing the dialogue structure and used the summaries of the transcripts.

[11] There was no overlap of annotators for the Pythagoras and MedForum dataset.

3.3.4 Dataset statistics

Table 3.2 lists characteristics of keyphrase assignment datasets including the German datasets *peDOCS*, *MedForum*, and *Pythagoras*. For comparison, the commonly used English datasets *Inspec* (Hulth, 2004), *DUC-2001* (Wan and Xiao, 2008a,b), and *SP* (Nguyen and Kan, 2007) are shown. We compare all six datasets with respect to three characteristics: (i) The type and language of documents, (ii) the size of the set of keyphrases, and (iii) the characteristics of the keyphrases in the datasets.

Of the six datasets, three are composed of scientific text. One contains full journal articles (SP), one paper abstracts (Inspec), and one a manifold spectrum of types including articles, full books, and reviews (peDOCS). Two further datasets are composed of text which can be expected to be of high quality, as they are news text (DUC-2001) or created by experts (Pythagoras). In comparison to that, MedForum is the only dataset which is composed of potentially noisy user-generated data. We divided the datasets in Table 3.2 in the German and English datasets. We will see that the language of the dataset has an influence on the characteristics.

Regarding the dataset size, we see that peDOCS and INSPEC are the largest in terms of the number of documents and Pythagoras is the smallest with only 60 documents. Inspec and peDOCS are by far the largest of the sets, since they have been created over the course of several years. Creating a huge dataset with keyphrase annotations is a tedious tasks, which explains the smaller size of all datasets with multiple annotators for each document. In the case of peDOCS and Inspec, each document was annotated by a single annotator, and the resulting dataset is used commercially or by official institutions. The other datasets are created for the purpose of evaluating keyphrase identification approaches.

The peDOCS dataset contains the longest documents on average, followed by SP. MedForum and Inspec are comparable in terms of document length as they contain forum posts and abstracts. The standard deviation of document length of MedForum and SP is relatively low compared to the other datasets because they were intentionally filtered by length. The peDOCS has the highest standard deviation of the document length, hence, making it a realistic dataset for a resource without any length requirements. The high deviation is due to few extremely long documents (i.e. dissertations that can easily cover several hundreds of pages).

The average number of keyphrases per document is comparable on all six datasets (ranging from 8.07 to 11.37). Documents in peDOCS have on average slightly more keyphrases (11.37 keyphrases). In the case of peDOCS, annotators were asked to add as many keyphrases as possible, while in the case of MedForum only the key aspects of a document should be extracted as keyphrases. There is no consistent correlation between the number of keyphrases and the document length for all datasets. In case of Inspec, this would be misleading, since in the annotation phase the whole paper was available, but only the abstract was included in the final dataset.

We see a major difference between German and English datasets in terms of the average number of tokens per keyphrase. Keyphrases in German datasets are on average very short (ranging from 1.07 to 1.30 tokens or 10.28 to 13.27 characters per keyphrase). In English datasets, the number of tokens per keyphrases are rather stable with 2.03 to 2.09 and 15.97 to 17.33 characters per keyphrase. This difference can be explored by consid-

[12] Document length in number of tokens.

Dataset	peDOCS	MedForum	Pythagoras	Inspec	DUC-2001	SP
Type	Articles, books, ...	Forum posts	Summaries	Paper abstracts	News articles	Science Papers
Language	German			English		
Number of documents	2,644	102	60	2,000	301	134
∅ document length[12]	14,016 ± 37,312	135 ± 13	277 ± 135	138 ± 67	902 ± 602	8,495 ± 2,008
Median document length[9]	809	104	68	22	167	1,903
Number of keyphrases	30,051	853	622	22,060	2,428	1,114
∅ keyphrases / document	11.37 ± 7.3	8.41 ± 2.7	10.37 ± 3.5	11.03 ± 5.7	8.07 ± 1.9	8.31 ± 2.5
∅ tokens / keyphrase	1.15 ± .44	1.07 ± .29	1.30 ± .62	2.03 ± .91	2.09 ± .74	2.08 ± .94
∅ characters / keyphrase	13.27 ± 5.7	10.28 ± 4.6	12.22 ± 5.7	17.33 ± 8.6	15.97 ± 6.4	17.15 ± 8.1

Table 3.2
Corpus statistics of keyphrase extraction datasets.

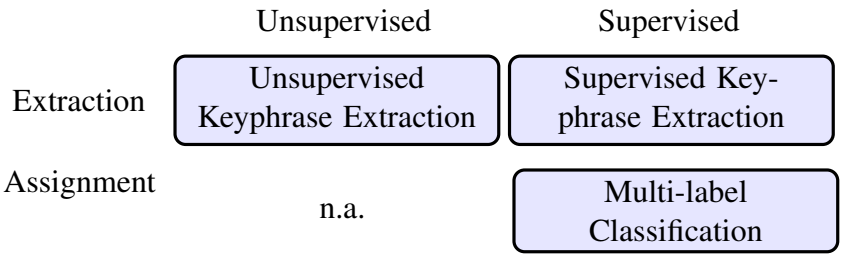


Figure 3.4: Overview of approaches to keyphrase assignment.

ering the potential keyphrase *Nachhilfelehrer* (Engl.: *private tutor*). It is a rather specific keyphrase which, however, consists of only one token in German, but two tokens in English (although the number of characters in this case is higher for German). Compounds are frequent as keyphrases in German. In English, keyphrases are rather noun phrases longer than a single token.

3.3.5 Dataset-specific user scenarios

We decided to use a rather broad selection of keyphrase datasets to cover the different scenarios described in Section 2.3 for German and English. The focused searcher (instrumental user type) is typically interested in highly-specific resources such as scientific papers (peDOCS, Inspec, SP) or specialized forums (MedForum). The recreational news reader would rely on news articles (DUC-2001) as a starting point. A knowledge worker in a company (instrumental user type, too) very often deals with an unorganized collection of diverse documents, best represented by the peDOCS dataset. In this scenario, the Pythagoras dataset represents a collection of summaries for meetings, e.g. business meetings. There is no one-to-one relation from datasets to scenarios as this is just an approximation of users, and the tasks are too manifold to be represented by a single dataset.

3.4 Approaches to Keyphrase Identification

Figure 3.4 follows the distinction of extraction and assignment approaches given in the previous section. In addition, we divide in unsupervised and supervised approaches. Unsupervised approaches do not require any further annotated data, while supervised approaches rely on existing data with annotated keyphrases.

We can extract keyphrases both in an unsupervised and supervised manner. For assignment, we apply multi-label classification based on training data with annotated keyphrases as labels. In contrast to supervised keyphrase extraction, the set of labels is predefined and cannot be extended by other phrases appearing in previously unseen documents. For supervised keyphrase extraction, the classification is based on a phrase's characteristics. For every candidate keyphrase, a set of features are extracted (Witten et al., 1999). Unsupervised approaches do not require any training data, while all supervised approaches consist of a training phase to create a model and a testing or evaluation phase, where the trained model is applied to new documents.

3.4.1 Keyphrase extraction

Keyphrase extraction approaches rank terms from the document according to a metric. They are based on phrases that appear in the document text. One very versatile metric for ranking terms, or in this case phrases, is the tf-idf[13] metric. For tf-idf, there exist different configurations, depending on the weighting of text and document frequency in the document collection.

In this section, we focus on approaches specifically applied to keyphrase extraction.

tf-idf

Salton and Buckley (1988) present the foundation of keyphrase extraction with the introduction of the tf-idf metric. The tf-idf metric relates the term frequency inside a document with the number of documents in the collection containing the term. Equation 3.1 shows the definition of tf-idf. In this formula $f(t, d)$ is the frequency of term t in document d, $|D|$ is the number of documents and $|d \in D : t \in d|$ is the number of documents mentioning term t.

$$\text{tf-idf}(t, d) = f(t, d) \cdot \log \frac{|D|}{|d \in D : t \in d|} \tag{3.1}$$

The term *tf* measures the importance of the phrase inside the current document; the term *idf* measures the *distinctiveness* of the term. Distinctive phrases are less frequently used in other documents. Phrases that appear frequently inside a document but infrequently in other documents of the collection have a high tf-idf value. Common phrases like stopwords (*and*, *but*) have a high frequency in all documents and thus have a lower tf-idf value. Phrases with a medium to high tf-idf value receive a higher score.

Tomokiyo and Hurst (2003) use language models to compute *informativeness*, i.e. terms with high information content and *phraseness*, i.e. terms that appear often as a multiword. Csomai and Mihalcea (2007) adopt this idea and use tf-idf values as a measure for *informativeness*. The phrase "Soccer World Cup" has high tf-idf values (*informativeness*) and the words are often used in this combination (*phraseness*), hence making it a good keyphrase.

Different configurations of the tf-idf metric have shown to perform better in different scenarios (Manning et al., 2008). As a modification to tf-idf, the inverse document frequency can be replaced with the text frequency in a background corpus D' (Rayson and Garside, 2000) such as the web (cf. Equation 3.2). The modified inverse document frequency is more like a background text frequency. Instead of counting the frequency in the corpus itself, the frequency in a—usually larger—background corpus[14] is computed. This has the advantage that no complete corpus needs to be processed in advance to apply this metric, or it can be applied to a single document. Results obtained are independent of the other documents in the collection.

Further modifications of tf-idf include Equation 3.3 and Equation 3.4. The metric in Equation 3.3 applies the inverse document frequency without taking the logarithm of the inverse document frequency. In Equation 3.4, the metric sets the inverse document

[13] The abbreviation describes the formula: text frequency times inverse document frequency

[14] In this thesis, we use Web1T (Brants and Franz, 2006) as the background corpus.

frequency to a constant value of 1. The latter configuration corresponds to text frequency.

$$\text{tf-idf}_{web}(t,d) = f(t,d) \cdot \log \sum_{d' \in D'} f(t,d') \tag{3.2}$$

$$\text{tf-idf}_{\text{normal}}(t,d) = f(t,d) \cdot \frac{|D|}{|d \in D : t \in d|} \tag{3.3}$$

$$\text{tf-idf}_{\text{constant}}(t,d) = f(t,d) \cdot 1 \tag{3.4}$$

tf-idf is one of the key approaches used in this thesis. We use different configurations of tf-idf in this thesis for keyphrase identification, table-of-contents generation, and link discovery.

Further approaches

Mihalcea and Tarau (2004a) introduce the unsupervised graph-based approach *TextRank* to extract keyphrases: A graph is created with keyphrase candidates as nodes. An edge is added if two keyphrase candidates co-occur in a certain context window (e.g. 3 words left or right of the anchor candidate, or in the same sentence as the anchor candidate) in the document. The weight of the edge is defined as the number of co-occurrences. The graph centrality measure PageRank (Page et al., 1999) is then used to rank the nodes in the graph. The highest ranked nodes are then selected as keyphrases. This approach is corpus-independent; no information from external resources is taken into account.

Supervised approaches are able to create a model for given training data and rely on a combination of metrics. In a training phase, a model specific to the dataset is learned and in the second phase, it is applied to new documents. Supervised approaches apply machine learning algorithms, e.g. decision trees (Turney, 2000) or Naïve Bayes (Witten et al., 1999), to solve this problem.[15]

For supervised approaches, tf-idf values and co-occurrence (TextRank) information can be used as features. The position of a candidate is also a good feature as a good keyphrase might be introduced early in the document. Machine learning allows for using many features, for which their importance is learned in a training phase. Using more features may further improve results. Part-of-speech information is valuable, as nouns are more likely keyphrases than prepositions (Hulth, 2003). Additionally, acronym identification techniques can be applied as they may also be good keyphrases (Nguyen and Kan, 2007).

Csomai and Mihalcea (2008) apply approaches from keyphrase extraction to the task of back-of-the-book indexing. They incorporate knowledge resources such as Wikipedia to compute the *keyphraseness*[16] of phrases. Hulth (2003) trains a supervised system for keyphrase extraction using linguistic features such as part-of-speech patterns. Supervised approaches often outperform unsupervised systems (Kim et al., 2010) for keyphrase extraction, as they can better capture characteristics of keyphrases in the dataset, e.g. using information about their length and part-of-speech tags.

[15] For an overview of machine learning algorithms, see (Kupietz and Belica, 2010).

[16] The keyphraseness is specified as how often a term is used as a keyphrase compared to the overall frequency.

Controlled vocabulary

An additional source of information is a controlled vocabulary, often referred to as a thesaurus. It is a list of previously collected terms which indexers often use for assigning keyphrases to documents. Medelyan and Witten (2006) state that the usage of a controlled vocabulary "eliminates the occurrence of meaningless or obviously incorrect phrases". Lopez and Romary (2010) use the existence of a term in domain-specific vocabulary as a feature. The thesaurus can be collected independently from the current collection as long as there is a high overlap of domains. This leads to many good keyphrases present in the thesaurus.

3.4.2 Keyphrase assignment

A controlled vocabulary is useful if there is an overlap between the terms in this vocabulary and the keyphrases in the document. In an extreme case, all keyphrases are present in this vocabulary. This might be due to an annotation policy, which allows only keyphrases from this vocabulary, or a procedure in which human annotators first try to select keyphrases provided by this vocabulary. In this case, keyphrases from the vocabulary can be automatically assigned, which can be done with multi-label classification. With multi-label classification, we assign one or more labels from a predefined label set to a document.

Documents can be clustered by their labels which allows creating overview pages and browsing through the collection (Jäschke and Marinho, 2007). These labels are similar to tags or categories (Sebastiani, 2002) and the set of all possible labels is also referred to as a tag set.[17] Instead of extracting keyphrases from the document only, any label from the label set can be assigned (Lipczak, 2008), thus performing keyphrase assignment.

Multi-label classification (Madjarov et al., 2012) first learns a classification model for the labels based on features. There are approaches for learning a model for the combination of labels[18] and approaches for learning a model for every label individually.[19] As an example, a classifier will most likely learn that a document containing the word *professor* should be assigned with the label *university education* if many training documents with the label *university education* contain the word *professor*. This requires that the classifier has a feature for the usage of the word *professor*. The document itself does not need to explicitly contain the phrase *university education*.

Figure 3.4 lists no approaches for unsupervised keyphrase assignment. However, there exist unsupervised document classification approaches which we chose not to present in this thesis, e.g. Slonim et al. (2002). Instead of assigning labels directly, the document collection is clustered based on the document texts. In the second step, topics for the clusters are assigned and can then be used as keyphrases (Blei et al., 2003).

3.4.3 Decompounding

Many approaches to automatic extraction of keyphrases are based on the assumption that frequent terms are more important to a document. To better compute these frequency

[17] In case of user-generated tags, they are often referred to as a folksonomy.

[18] So called ensemble methods (Tsoumakas et al., 2011).

[19] Referred to as binary relevance method (Tsoumakas and Katakis, 2007; Read et al., 2011).

counts, we apply some kind of normalization, e.g. lemmatization or noun chunking (Hulth, 2003; Mihalcea and Tarau, 2004a), in order to arrive with more accurate counts. However, especially in German the frequent use of noun compounds has an adverse effect on the reliability of frequency counts. Consider for example a German document that talks about *Lehrer* (Engl.: *teacher*) without ever mentioning the word *Lehrer* at all, because it is always used inside compounds like *Nachhilfelehrer* (Engl.: *private teacher*) or *Gymnasiallehrer* (Engl.: *grammar school teacher*). Thus, we argue that we can rely on more accurate frequency counts by splitting compounds in meaningful parts, i.e. by performing decompounding. We introduce approaches for decompounding in the following section.

The benefit of decompounding has been shown in previous work. Koehn and Knight (2003) improve machine translation on German-English by using a frequency based decompounding approach. Although this is not the most accurate approach according to Koehn and Knight (2003)'s experiments for decompounding German compounds, it produces the best outcome for the German-English translation task. Baroni et al. (2001) report an improvement in terms of keystroke saving rate for predicting the next words and characters. Hollink et al. (2004) apply decompounding to improve information retrieval in selected European languages (not including English) and report significant improvement for German and Swedish. Ordelman (2003) is able to improve the results for Dutch speech recognition, by using a data-driven decompounding algorithm. To the best of our knowledge, we are the first to examine the influence of decompounding on keyphrase extraction.

3.5 Dealing with Compounds

Compounds are words which are composed of at least two other words and the meaning of the compound is given by the meaning of its parts. The German word *Abendessen* (Engl.: *dinner*) consists of *Abend* (Engl.: *evening*) and *Essen* (Engl.: *food*). In many languages, compounds make up a large proportion of the overall number of types. Based on an investigation of the APA corpus,[20] about half (47%) of the types in German are compounds (7% of the tokens) (Baroni et al., 2001). We differentiate here between tokens and types. The proportion of compounds in the overall number of tokens is much lower, because they are less frequently used. However, most of these compounds appear infrequently with 83% appearing less than five times. Schiller (2006) identified 43% of 420,000 types from German newspaper articles as compounds (5.5% of 9.3 million tokens). Depending on the text type, the ratio of compounds vary. In a manual for printers, 12% of all tokens were compounds (Schiller, 2006).

In some languages, e.g German, it is possible to construct words of unlimited length. The famous example *Donaudampschiffahrtskapitän...* (Engl.: *captain of a steam boat on the river Danube ...*) can easily be further extended. Compounds are convenient for humans to express their thoughts, but they impose a challenge for automatic text processing. They are hard to split automatically and even sometimes cannot be split accurately without any context. The German word *Wachstube* is ambiguous and can, depending on the context, be split as *Wach-stube* (Engl.: *guard room*), or as *Wachs-tube* (Engl.: *wax tube*).

[20] The corpus of the Austria Presse Agentur (APA) consists of 28 million words.

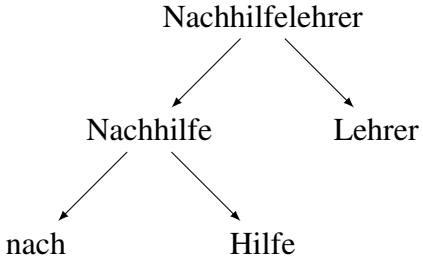


Figure 3.5: Decompounding of German term *Nachhilfelehrer* (Engl.: *private tutor*).

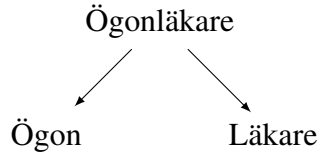


Figure 3.6: Decompounding of Swedish term *Ögonläkare* (Eng: eye doctor).

Figure 3.5 shows an example of a German compound, which can be split in a hierarchy. A hierarchy can be constructed by splitting compounds recursively. It contains the words *Nachhilfe* (Engl.: *private lesson*) and *Lehrer* (Engl.: *teacher*). *Nachhilfe* can again be split in *Nach* (Engl.: ∅[21]) and *Hilfe* (Engl.: *help*). Instead of processing the word *Nachhilfelehrer*, an automatic system can make use of the identified compound parts, which allows searching for any mentions of *Lehrer* and still finding the compound. Compounding is not specific for German, but exists in several germanic languages like Swedish as shown in Figure 3.6. The Swedish word *Ögonläkare* (Engl.: *eye doctor*) consists of the words *Ögon* (Engl.: *eye*) and *Läkare* (Engl.: *doctor*).

3.5.1 Approaches to compound splitting

Decompounding is usually performed in two steps: (i) a splitting algorithm creates candidates, and (ii) a ranking function decides which candidates are best suited for splitting the compound. For example, *Aktionsplan* has two splitting candidates: *Aktion(s)+plan* (Engl.: *action plan*) and *Akt+ion(s)+plan* (Engl.: *nude ion plan*).[22] After generating the candidates, the ranking function assigns a score to each splitting candidate, including the original compound. The original compound must be included, because in some cases, a compound has such strong inherent semantics that splitting it would not be reasonable. This is the case for *Deutschland* (Engl.: *Germany*), for which a splitting algorithm might generate the candidates *Deutsch* (Engl.: *German*) and *Land* (Engl.: *country*). In this case, a good ranking function would assign a higher score to *Deutschland* than to *Deutsch+land*. We will now take a closer look on possible splitting algorithms and ranking functions.

[21] A prefix – without sensible translation into English in this context.
[22] The additional 's' is a linking morpheme (Langer, 1998)

Splitting algorithms

Left-to-Right grows a window over the input from left to right. When a word from a dictionary is found a split is generated. The algorithm is then applied recursively to the rest of the input. The stopping criterion is when no further words are found. Among the splitting approaches, this is the one which generates the most candidates.

JWord Splitter[23] performs a dictionary look-up from left to right, but continues this process if the remainder of the word cannot be found in the dictionary. After it finds words in both parts (left and right), it creates a split and stops. If there is more than one possibility to split the compound, then the split which goes furthest to the right is the generated candidate. For instance, *Grundschullehrer* (Engl.: *elementary school teacher*) would generate the candidate *Grundschul(e) + Lehrer* (Engl.: *elementary school + teacher*).

Banana Splitter[24] searches for the word from the right to the left, and if there is more than one possibility, the one with the longest split on the right side is taken as candidate. For instance, *Grundschullehrer* would generate the candidate *Grund + Schullehrer* (Engl.: *basic + school teacher*).

Data Driven counts the number of words in a dictionary, which contain a split at this position as prefix or suffix for every position in the input. A split is made at the position with the largest difference between prefix and suffix counts (Larson and Willett, 2000).

ASV Toolbox[25] uses a trained Compact Patricia Tree to recursively split parts from the beginning and end of the word (Biemann et al., 2008). Unlike the other algorithms, it generates only a single split candidate at each recursive step. For that reason, it does not need a ranker. It is also the only supervised (using lists of existing compounds) approach tested.

Ranking functions

As stated earlier, the ranking functions are as important as the splitting algorithms, since a ranking function is responsible for assigning scores to each possible decompounding candidate. For the ranking functions, Alfonseca et al. (2008) use a geometric mean of unigram frequencies (Equation 3.5), and a mutual information function (Equation 3.6).

$$r_{Freq}() = \left(\prod_{i}^{N} f(w_i) \right)^{\frac{1}{N}} \tag{3.5}$$

$$r_{M.I.}() = \begin{cases} -f(c) \log f(c) & \text{if } N = 1 \\ \frac{1}{N-1} \sum_{i}^{N-1} \log \frac{bigr(w_i, w_{i+1})}{f(w_i) f(w_{i+1})} & \text{otherwise} \end{cases} \tag{3.6}$$

[23] `https://github.com/danielnaber/jwordsplitter` (last accessed: 2014-12-07)
[24] `http://niels.drni.de/s9y/pages/bananasplit.html` (last accessed: 2014-12-07)

In these equations, N is the number of fragments the candidate has, w is the fragment itself, $f(w)$ is the relative unigram frequency for that fragment w, $bigr(w_i, w_j)$ is the relative bigram frequency for the fragment w_i and w_j, c is the compound itself without being split.

The geometric mean as described in Equation 3.5, assigns a lower rank to split candidates with one very unlikely fragment. An unknown fragment has the frequency 0 and thus, the split candidate receives the lowest score. Using mutual information as in Equation 3.6, incorporates probabilities that certain fragments appear as bigrams in a background corpus. The higher the average frequency of a bigram compared to their unigram frequency is, the higher the candidate is ranked.

3.5.2 Decompounding Experiments

The corpus created by Marek (2006) is used as a gold standard to evaluate the performance of the decompounding methods. This corpus contains a list of 158,653 compounds, stating how each compound should be decompounded. The compounds were obtained from the issues 01/2000 to 13/2004 of the German computer magazine c't[26] in a semi-automatic approach. Human annotators reviewed the list to identify and correct possible errors. This dataset is extensive, however, it does not contain any non-compounds. We use the igerman98 dictionary[27]. For calculating the required frequencies, we used the Web1T 5-gram corpus (Brants and Franz, 2006).

Koehn and Knight (2003) use a modified version of precision and recall for evaluating decompounding performance. We decided to apply these metrics for measuring the splitting algorithms, and ranking the functions' performance. The following counts were used for evaluating the experiments on the compound level: *correct compound (cc)*, a compound which was correctly split (including all fragments); *wrong faulty compound (wfc)*, a compound which was wrongly split; *wrong non compound (wnc)*, a compound which was not split. The adaptation for decompounding is necessary because there is not only a division in relevant and irrelevant, but a compound can also be wrongly split.

Furthermore, we also count on a split level: *correct split (cs)*, a split fragment which was correctly identified; *wrong split (ws)*, a split fragment which was wrongly identified.

P_{comp} and R_{comp} evaluate decompounding on the level of compounds, and we propose P_{split} to evaluate on the level of splits. There are no non-compounds in the dataset, which eliminates any *false positives*. One could also consider any missed compound as a wrong split, thus adding *wfc* and *wnc*.

$$P_{comp} = \frac{cc}{cc + wfc} \tag{3.7}$$

$$R_{comp} = \frac{cc}{cc + wfc + wnc} \tag{3.8}$$

$$P_{split} = \frac{cs}{cs + ws} \tag{3.9}$$

[26] `http://www.heise.de/ct/` (last accessed: 2014-12-07)

[27] https://www.j3e.de/ispell/igerman98/index_en.html

Splitter	Ranker	P_{comp}	R_{comp}	P_{split}
Banana	Freq.	.70	.40	.83
	M.I.	.66	.16	.81
Data Driven	Freq.	.49	.18	.70
	M.I.	.40	.04	.58
JWord	Freq.	.67	.63	.79
	M.I.	.59	.20	.73
Left-to-right	Freq.	.64	.58	.71
	M.I.	.26	.08	.33
ASV ToolBox		**.80**	**.75**	**.87**

Table 3.3
Evaluation results of state of the art decompounding systems.

As we focus in this work on the influence of decompounding on improving the accuracy of frequency counts, P_{split} is the best metric in our case. For instance, *Semesterwochenstunde* (Engl.: *weekly hours per semester*) is correctly split as *Semester + Woche(n) + Stunde*. If it is split like *Semester+Wochenstunde*, at least one split fragment is correct, still influencing the frequency counts. We can see in Table 3.3 that the ASV Toolbox splitting algorithm is the best performing system in respect to P_{split}.

In this section, we have analyzed decompounding as an extension to frequency-based keyphrase extraction approaches. We have shown that we can achieve high precision and recall for decompounding and can use decompounding for text structuring with keyphrases.

3.6 Experimental Setup

In this section, we describe the setup for the keyphrase identification experiments, including evaluation metrics, datasets, and approaches.

3.6.1 Evaluation Metrics

For the keyphrase experiments, we compare results in terms of precision and recall of the top-5 keyphrases (P@5), Mean Average Precision (MAP), and R-precision (R-p).[28] Precision@5 is the ratio of true positives in the set of extracted keyphrases when five keyphrases are extracted. Recall@5 is the ratio of true positives in the set of gold keyphrases when five keyphrases are extracted. MAP is the average precision of extracted keyphrases from 1 to the number of extracted keyphrases, which can be much higher than ten. R-precision[29] is the ratio of true positives in the set of extracted keyphrases when as many

[28]Using the top-5 keyphrases reflects best the average number of keyphrases in our evaluation datasets (between 8.07 and 11.37) and is common practice in related work (Kim et al., 2013).

[29]This is a commonly used measure in information retrieval and first used for keyphrase identification in Zesch (2009)

keyphrases as there are gold keyphrases are extracted. R-precision focuses more on the ranking of keyphrase candidates if less than ten gold keyphrases exist.[30]

3.6.2 Datasets

We use the datasets presented in Section 3.3 for our experiments. We primarily focus on the dataset peDOCS, which we presented in earlier work (Erbs et al., 2013a). This dataset has a large number of documents (2,644) and keyphrases are annotated by experts. As the peDOCS dataset contains German documents, we can compare results for German and English datasets (even when having different properties). For comparison with English documents, we rely on Inspec, DUC-2001, and SP.

We further use two newly created domain-specific German datasets: (i) MedForum contains forum posts from the medical domain, and (ii) Pythagoras consists of lesson summaries. We use these datasets to evaluate the benefit of decompounding for keyphrase identification.

3.6.3 Preprocessing

For preprocessing, we rely on components from the DKPro Core framework (Eckart de Castilho and Gurevych, 2014) and on DKPro Lab (Eckart de Castilho and Gurevych, 2011) for building experimental pipelines. We use the Stanford Segmenter[31] for tokenization, TreeTagger (Schmid, 1994) for lemmatization and part-of-speech tagging of German text. For English text, we use tagging and lemmatization (Toutanova and Manning, 2000; Toutanova and Klein, 2003) and named entity recognition (Finkel et al., 2005) components from Stanford Core NLP (Manning et al., 2014). Finally, we perform stopword removal and decompounding as described in Section 3.5.

It should be noted that in most preprocessing pipelines, decompounding is the last step, as it heavily influences part-of-speech tagging. Consider for example the noun compound *Nachhilfe*, which is split into the prefix *Nach* and the noun *Hilfe*. Performing part-of-speech tagging after decompounding would yield inferior results, as the noun is replaced with a preposition-noun combination that cannot be properly analyzed by the part-of-speech tagger.

3.6.4 Approaches to keyphrase identification

We divide the selected approaches into categories to allow for an isolated evaluation of their effect on keyphrase identification. We follow the categories described in Section 3.4.

Keyphrase extraction

We extract all lemmas in the documents as keyphrase candidates and rank them based on frequency counts and their position in the document. We also experiment with other keyphrase candidates, such as noun phrases, tokens, and named entities.

[30] Refer to Buckley and Voorhees (2000) for an overview of evaluation measures and their characteristics.
[31] http://nlp.stanford.edu/software/segmenter.shtml

It has been shown that position and frequency heuristics are very strong baselines (Zesch, 2009). We thus use the position of a phrase[32] and the text frequency as baselines in our experiments. We normalize by the number of phrases. We compute upper bounds based on keyphrases assigned by the expert annotators. If two or more annotators identified gold keyphrases for a dataset, we measure the average performance of one annotator compared to the remaining annotators in terms of the evaluation metrics.

We evaluate the following ranking methods: tf-idf, tf-idf$_{web}$ with different weighting strategies as described in Section 3.4.1. We further evaluate the graph-based approach TextRank (Mihalcea and Tarau, 2004a).

Controlled vocabulary Previous work (Medelyan and Witten, 2006; Lopez and Romary, 2010) states that domain-specific or controlled vocabularies further improve the performance of keyphrase extraction. Thus, we use the described thesauri for peDOCS (c.f. Table 3.1) as a filter for our extracted keyphrases. This filter checks whether an identified keyphrase appears in the thesaurus and rejects it if it is not included. Only keyphrases which are included in the thesaurus are accepted. This limits keyphrases to a predefined set of keyphrases, but the score for each keyphrase remains unchanged.[33]

Keyphrase assignment

As shown in Figure 3.3, keyphrases are not equally distributed. Some keyphrases are used only once, while few are used very frequently. These frequently used keyphrases can be assigned with multi-label classification. Instead of using a thesaurus, we use the most frequent keyphrases as labels for classification. We introduce the parameter n as the size of our label set. More keyphrases can be covered if n is set to a higher value but on average fewer examples will be available for each label. Examples are documents for which an indexer has assigned the corresponding labels. Classification algorithms require positive (documents with a specific label) and negative (documents without this label) examples to learn a model. In case of labels with few positive examples (documents with label), the training data is not sufficient to train a reliable model.[34]

We evaluate results for the multi-label classification approach under identical conditions as done for keyphrase extraction. We compare classified labels to manually assigned keyphrases and measure results in terms of precision, recall, and R-precision.[35] We use the open source software tool Mulan (Tsoumakas et al., 2010) based on WEKA (Hall et al., 2009) and apply cross-validation to avoid leaking information from the learning to the evaluation phase. We use the top-500 most frequent n-grams[36] from the dataset as features. We use two frequently used classification approaches: support vector machines

[32]The closer the keyphrase is to the beginning of the text, the higher it is ranked. This is not dependent on frequency counts, but decompounding can also have an influence if a compound that appears early in the document is split into parts that are now also possible keyphrase candidates.

[33]In contrast to keyphrase assignment, no training data is required for filtering with a controlled vocabulary.

[34]There is no fixed number of positive examples required for classification, but with increasing sample size the probability of the model is better increases (Beleites et al., 2013).

[35]We do not report the accuracy for each of the labels, as we are interested in the overall performance for keyphrase assignment.

[36]We use unigrams, bigrams and trigrams.

Approach	Precision@5	Recall@5	R-precision	MAP
Upper bound	.856	.393	.614	.614
Position (baseline)	.096	.042	.092	.083
tf-idf$_{\text{constant}}$	.170	.075	.127	.123
tf-idf	.148	.065	.115	.112
tf-idf$_{\text{normal}}$	.006	.003	.005	.009
tf-idf$_{web}$	**.188**	**.083**	**.139**	**.139**
TextRank	.153	.067	.112	.116

Table 3.4
Results for keyphrase extraction approaches on peDOCS using lemmas as potential keyphrases. Best results are marked bold.

(SVM)[37] (Cortes and Vapnik, 1995) and decision trees (J48)[38] (Quinlan, 1992).

Decompounding

We test each of the keyphrase ranking methods with (w) and without (w/o) decompounding. For our evaluation, we could not rely on English datasets, as they contain only few compounds and thus the expected effect of decompounding is small. German is a good choice, as it uses compounds extensively.

3.7 Experimental Results and Discussion

In this section, we describe and discuss experimental results obtained with different approaches based on the categories described in Section 3.4.

3.7.1 Keyphrase Extraction Experiments

We first analyze results on the largest German dataset. peDOCS contains 2,644 documents of variable length (see Section 3.3.2) and is an example of dataset in the educational domain. We select all lemmas as potential keyphrases and filter these by their part-of-speech. A preliminary analysis has shown that nouns and adjectives yield best results. We lemmatize[39] extracted and manually assigned keyphrases to map different forms of the same word to the base form.

Table 3.4 displays the results for keyphrase extraction on the peDOCS dataset. We provide information for the upper bound for precision and recall among the top five keyphrases, R-precision, and mean average precision (MAP). The upper bound for precision@5 is limited to .856 because not for every document, there are five manually assigned keyphrases. It is further reduced by some manually assigned keyphrases not contained in the document text and thus impossible to capture by extraction approaches. The upper bound for recall@5 is limited by documents having more than five manually

[37] Using RakEL (Tsoumakas et al., 2011) as meta algorithm.
[38] Using BRkNN (Spyromitros et al., 2008) as meta algorithm.
[39] As stated in section 3.6.3, we use the TreeTagger trained on the German data (Schmid, 1994).

assigned keyphrases. The upper bound for R-precision and mean average precision is limited by the ratio of keyphrases appearing in the document.

We compare keyphrase extraction approaches to the position baseline (the earlier in the document a phrase appears, the higher it is ranked). Many scientific documents include an abstract, containing many keyphrases, thus making the position a strong baseline. We present results for different configurations for the tf-idf metric, and TextRank as another state of the art approach (cf. Section 3.4.1). Our evaluation of different modifications of tf-idf (see Section 3.4.1 for details) shows that using the tf-idf$_{web}$ as defined in Equation 3.2 performs best across all evaluation metrics. tf-idf$_{\text{constant}}$ (see Equation 3.4) yields second best results.

Analyzing the difference with respect to precision and R-precision, we see that overall results for R-precision are lower than for precision. This is due to the different number of extracted keyphrases considered for evaluation. With precision, the number of extracted keyphrases is fixed (in our case 5) and with R-precision, it is dependent on the number of gold keyphrases (equal number of gold and extracted keyphrases). Lower values for R-precision indicate that the higher ranked keyphrases are indeed more likely gold keyphrases (manually assigned keyphrases).

tf-idf$_{web}$ and tf-idf$_{\text{constant}}$ perform better than tf-idf. The weaker performance of classic tf-idf might be due to many keyphrases that are used throughout the dataset, e.g. *Deutschland* and *Bildung*, which lead to a high document frequency and thus to a low tf-idf value. This is an issue of the domain-specific nature of the peDOCS dataset. The document collection is not a reliable source for document frequencies as most of the documents share a set of keyphrases. tf-idf$_{web}$ uses document frequencies from another source—in this case the web through the Web1T corpus—and can thus provide more reliable document frequencies. tf-idf$_{\text{constant}}$ sets the normalization factor to 1, hence solely relying on the text frequency. tf-idf$_{\text{normal}}$ does not use any normalization for the document frequency. This emphasizes the issue of having many documents containing many keyphrases. Even worse, terms are not normalized by the logarithm of their frequency, but just by their frequency. This results in most frequently used keyphrases to be ranked very low. TextRank yields results comparable to tf-idf$_{web}$ in terms of precision and recall, but lower results in terms of R-precision. TextRank creates a graph representation of the document based on co-occurrences and does not normalize scores in any way.

The overall results are rather low, but they confirm state of the art results in Hasan and Ng (2010). Further, computing precision and recall within the first five keyphrases leads to a low upper bound for recall (.393). A manual error analysis revealed that the task of keyphrase extraction is indeed a very hard one. Many extracted keyphrases are only a partial match to manually assigned keyphrases, other extracted keyphrases are semantically similar to manually assigned keyphrases, and further extracted keyphrases might not be included in the list of manually assigned keyphrases, but nevertheless be valuable keyphrases. Our manual error analysis shows all three cases of misses of extracted keyphrases. To investigate these issues further, we compare results on peDOCS with results on English data.

Comparison to results on English data

Table 3.5 compares results on the German dataset peDOCS with the three English datasets INSPEC, DUC, and SP. It shows results obtained with TextRank (Mihalcea and Tarau,

Dataset	Approach	Precision@5	Recall@5	R-precision	MAP
peDOCS	*Upper bound*	.856	.393	.614	.614
	Position (baseline)	.096	.042	.092	.083
	tf-idf$_{\text{constant}}$	.170	.075	.127	.123
	tf-idf	.137	.060	.107	.112
	tf-idf$_{web}$	**.188**	**.083**	**.139**	**.139**
	TextRank	.153	.067	.112	.116
Inspec	*Upper bound*	.912	.409	.690	.690
	Position (baseline)	.140	.063	.130	.175
	tf-idf$_{\text{constant}}$	.146	.065	.134	.170
	tf-idf	.160	.071	.154	**.190**
	tf-idf$_{web}$	**.169**	**.075**	**.153**	.188
	TextRank	.140	.062	.128	.166
DUC-2001	*Upper bound*	.990	.555	.851	.851
	Position (baseline)	.084	.053	.073	.090
	tf-idf$_{\text{constant}}$	.102	.065	.088	.092
	tf-idf	**.154**	**.097**	**.129**	**.139**
	tf-idf$_{web}$	.116	.073	.103	.106
	TextRank	.092	.058	.077	.081
SP	*Upper bound*	.949	.538	.808	.808
	Position (baseline)	**.107**	**.064**	.079	.095
	tf-idf$_{\text{constant}}$	.008	.048	.073	.089
	tf-idf	.071	.043	.063	.064
	tf-idf$_{web}$	.102	.061	**.094**	**.104**
	TextRank	.080	.048	.073	.084

Table 3.5
Results of unsupervised keyphrase extraction across all datasets. We use lemmas for peDOCS and n-grams (with $n \leq 3$) for the English datasets.

2004a) and the various configurations of the tf-idf approach. For the peDOCS dataset, we have previously shown that tf-idf$_{web}$ yields best results, followed by tf-idf$_{\text{constant}}$ and tf-idf.

For the Inspec dataset, tf-idf$_{web}$ yields best results across all evaluation metrics, outperforming tf-idf$_{\text{constant}}$ and tf-idf. Again, TextRank performs slightly worse than the tf-idf configurations, but still outperforms the position baseline. The Inspec dataset contains scientific abstracts of few domains, making it a closed domain dataset. As in the peDOCS dataset, having a closed domain dataset is an issue for the classic tf-idf configuration. Domain-specific terms are lower ranked because they are normalized by the document frequency.

For the DUC-2001 dataset, tf-idf yields best results, while tf-idf$_{web}$ yields second best results. The dataset consists of news articles covering multiple topics. Thus, tf-idf performs better, as normalization (the idf-term) has a positive effect on the ranking of keyphrases. The position baseline yields the worst results in terms of precision@5 among the English datasets. Both other English datasets comprise scientific text, only the

DUC-2001 dataset comprises news text. Non-scientific text apparently does not start with mentioning keyphrases.

Only for the SP dataset, the position baseline yields best results in terms of precision and recall with five extracted keyphrases. They slightly outperform tf-idf$_{web}$, which yields best results in terms of R-precision and mean average precision. The SP dataset consists of full scientific papers, thus starting with an abstract containing many keyphrases. The position baseline benefits from this. tf-idf$_{web}$ performs better than the position baseline in terms of R-precision and mean average precision. Within the five highest ranked keyphrases, the position baseline performs better, while the remaining keyphrases are ranked worse. A scientific paper starts with an abstract containing many keyphrases, but then continues with a broader introduction containing fewer keyphrases. After extracting the first few keyphrases with a position approach, tf-idf$_{web}$ yields better results. Surprisingly, tf-idf$_{\text{constant}}$ yields very poor results for this dataset, as opposed to very good results for peDOCS. This is due to the different selection of candidates for keyphrases. For English datasets, n-grams (with $n \leq 3$) yield best results, but weighting n-grams by their text frequency yields bad results. Common, yet unimportant, n-grams are weighted highest. As all n-grams with $n \leq 3$ are potential keyphrases, all frequent words[40] are ranked highest.

The ranking of the keyphrase extraction approaches is rather stable across the datasets A direct comparison of the results across languages is hard because preprocessing (e.g. POS tagging) for different languages does not perform equally well. Overall, results are very low which might be due to skipping filtering techniques[41] and keeping manually assigned keyphrases which do not appear in the document as part of the gold standard. tf-idf yields best results on the open domain datasets, while tf-idf$_{web}$ yields better results for closed domain datasets. Approaches yield higher results for datasets with shorter documents.

Controlled vocabulary

Table 3.6 shows results of keyphrase extraction approaches on the peDOCS dataset. As described in Section 3.3.2, professional indexers for the peDOCS dataset had two controlled vocabularies available. The controlled vocabularies help them to select keyphrases for a document, however they are not restricted to these vocabularies. The extended list contains 8,487 keyphrases and covers 49.9% of the used keyphrases. The core list is much smaller with 974 keyphrases and covers only 26.3% of the used keyphrases. As shown in Table 3.6, reducing the number of potential keyphrases also decreases the upper bound. For the core list, the upper bound for precision@5 is reduced to .552.

Without using any filter[42], tf-idf$_{web}$ yields highest results with a R-precision of .139. With both filters, tf-idf$_{web}$ yields even better results (.201 with the extended list and .175 with the core list), however, it is outperformed by classic tf-idf. Using the extended list improves results of all approaches across all evaluation metrics. Using the even smaller core list does not improve results any further for all approaches. Only the position baseline yields better results in terms of R-precision. Additionally, results in terms of mean average

[40] At least those that are not filtered out by our stopword filter.

[41] See Kim et al. (2013) for an overview of filtering techniques.

[42] We perform a look-up of the lemma of the extracted keyphrase and remove it if it is not included in the vocabulary.

Vocabulary	Approach	Precision@5	Recall@5	R-precision	MAP
None (30,051 keyphrases)	*Upper bound*	.856	.393	.614	.614
	Position (baseline)	.096	.042	.092	.083
	tf-idf$_{\text{constant}}$	.170	.075	.127	.123
	tf-idf	.137	.060	.107	.112
	tf-idf$_{web}$	**.188**	**.083**	**.139**	**.139**
	TextRank	.153	.067	.112	.116
Extended list (8,487 keyphrases)	*Upper bound*	.793	.322	.499	.499
	Position (baseline)	.195	.086	.127	.118
	tf-idf$_{\text{constant}}$	.269	.118	.192	.181
	tf-idf	**.298**	**.131**	**.211**	**.197**
	tf-idf$_{web}$	.282	.124	.201	.188
	TextRank	.255	.112	.182	.173
Core list (974 keyphrases)	*Upper bound*	.552	.170	.263	.263
	Position (baseline)	.168	.074	.137	.163
	tf-idf$_{\text{constant}}$	.258	.113	.170	.185
	tf-idf	**.279**	**.123**	**.181**	**.194**
	tf-idf$_{web}$	.268	.118	.175	.190
	TextRank	.252	.111	.166	.180

Table 3.6
Results of keyphrase extraction approaches using a controlled vocabulary for the peDOCS dataset.

precision improve for all approaches but tf-idf.

The greater improvement of tf-idf compared to tf-idf$_{web}$ when using a controlled vocabulary is due to a better ranking of the remaining keyphrases. Without any controlled vocabulary, tf-idf ranks many irrelevant terms higher because their document frequency is low. These irrelevant terms are not included in the controlled vocabulary and thus filtered out.

3.7.2 Multi-label Classification Experiments

So far, we have investigated keyphrase extraction approaches. These approaches have an upper bound, because keyphrases not included in the document text cannot be extracted. For the peDOCS dataset, only 61% of all keyphrases are contained in the document text. We thus apply multi-label classification on the peDOCS dataset.

Table 3.7 shows evaluation results for multi-label classification depending on the label set size n. The label set is constructed by taking the n most frequent keyphrases as labels.[43] The size of the label set determines the upper bound for assigning keyphrases. The upper bound for recall@5 is higher the more labels are used because only those keyphrases which are used as labels can be extracted. However, using more labels decreases precision@5, because there is less training data available for less frequent labels. Extending the label set allows a higher recall (increase from .113 for 10 labels to .316 for

[43]The frequency of a label is counted in the development set, which was kept separate from the remaining dataset.

Label set n	Algorithm	Precision@5	Recall@5	R-prec.
10	*Upper bound*	.969	.113	.113
	J48	.358	.043	.058
	SVM	**.501**	.032	.051
20	*Upper bound*	.969	.154	.154
	J48	.309	.055	.066
	SVM	.406	.039	.055
50	*Upper bound*	.969	.234	.234
	J48	.305	**.063**	.065
	SVM	.337	.050	.058
200	*Upper bound*	.969	.316	.316
	J48	.330	.061	**.066**
	SVM	.322	.060	.063

Table 3.7
Results for multi-label classification approaches for peDOCS dataset.

200 labels). However, precision decreases for a larger label set size, especially, in case of SVM for which precision drops from .501 (10 labels) to .322 (200 labels). A label set size of 200 is a good trade-off between precision and recall. Although, larger label sets are possible, we limit the label set size to 200 as computation time increases with size.[44] We also omit reporting results for mean average precision as multi-label classification returns only few keyphrases, thus making precision, recall, and R-precision better suitable.

The classification algorithms J48 and SVM perform almost on par. J48 provides better results in terms of recall and R-precision, while SVM performs better in terms of precision. For label set size of 10, using SVM reaches the best results in terms of precision (.501).

Overall, results in Table 3.7 show that multi-label classification assigns keyphrases (labels) with higher precision but lower recall compared to keyphrase extraction. The low results in terms of recall and R-precision are due to a lower number of classified labels: Recall is limited if less than ten labels are classified. However, in order to effectively use multi-label classification, the dataset needs to be from a closed domain. In the peDOCS dataset, many documents share common keyphrases, which can then be used as labels. In open domain datasets, fewer keyphrases are shared among many documents, hence, the upper bound for different label set sizes will decrease.

3.7.3 Manual error analysis on peDOCS

Previously, we analyzed keyphrase extraction and multi-label classification approaches. We have seen that both approaches have their advantages and their shortcomings. Extraction approaches yield better results in terms of recall, while multi-label classification approaches yield better results in terms of precision. We will now investigate this with one example document.

[44] 10-fold cross-validation takes about 18 hours on a workstation with quad-core processor.

Manually assigned keyphrases	Keyphrase extraction (tf-idf$_{web}$)	Multi-label classification (J48, 200 labels)
Bildungsforschung	Promotion	**Statistik** (1.0)
Wissenschaft	Schulbereich	Bildungspolitik (0.75)
Hochschullehrerin	**Wissenschaft**	Schule (0.40)
Berufung	Männer	Berufsbildung (0.33)
Professur	Frauenanteil	
Hochschule	**Frau**	
Forschung	**Hochschule**	
Frau	Kulturwissenschaft	
Bildungsstatistik	Hausberufung	
Statistik	Deutschland	
...	...	

Table 3.8
Manually and automatically identified keyphrases for an example document. Correctly assigned keyphrases are marked bold, and scores for multi-label classification are given in parentheses. The first five extracted and classified keyphrases are separated to show the cut-off for computing precision@5 and recall@5.

Table 3.8 provides manually assigned keyphrases for an example document[45] and keyphrases assigned with supervised keyphrase extraction and multi-label classification. In total, 21 keyphrases are manually assigned to this documents (we listed the first ten). Some keyphrases are very similar, e.g. *Statistik* (Engl.: *statistics*) is the general term for *Bildungsstatistik* (Engl.: *educational statistics*). Keyphrase extraction returns a weighted list which is cut-off after ten keyphrases. Multi-label classification only assigns four keyphrases to this document.

Keyphrase extraction successfully identifies three keyphrases within the top-10 list and multi-label classification correctly assigns the highest ranked keyphrase. We observe several near misses[46] of keyphrases, e.g. the extracted keyphrase *Hausberufung* (Engl.: *internal appointment*) is a near miss for the gold keyphrase *Berufung* (Engl.: *appointment*). Additionally, we observe that most of the extracted and classified keyphrases not appearing in the list of keyphrases, are still good keyphrases for the document, e.g. *Frauenanteil* (Engl.: *percentage of women*) and *Bildungspolitik* (Engl.: *education policy*). Hence, the list of gold keyphrases cannot be considered complete and further identified keyphrases are not necessarily incorrect.[47] Rather than counting matches of keyphrases, we believe that an extrinsic evaluation of their usefulness could provide better insights into the system's quality.

In the list of extracted keyphrases, we see a special case of near misses: *Hausberufung* (Engl.: *internal appointment*) is a compound, having as one part the gold keyphrase *Berufung* (Engl.: *appointment*). In the following, we analyze the effect of a prior decompounding and show that it influences results on a German dataset.

[45] Title: *Chancengleichheit in Wissenschaft und Forschung* (Engl.: *Equal Opportunities in Science and Research*)

[46] Near misses are identified keyphrases which partially cover or are covered by a gold keyphrase in terms of meaning.

[47] This has further implications on the creation of datasets for keyphrase identification.

Method	Δ precision@5	Δ recall@5	Δ R-precision	Δ MAP
Position (baseline)	.000	.000	.000	.000
tf-idf$_{\text{constant}}$	.039	.030	.022	.012
tf-idf	.031	.024	.025	.015
tf-idf$_{web}$	.035	.021	.024	.012
TextRank	.000	.000	.000	.000

Table 3.9
Improvement with decompounding on the MedForum dataset with all lemmas as potential keyphrases.

3.7.4 Decompounding

In order to assess the influence of decompounding on keyphrase extraction, we evaluate the selected extraction approaches with (w/) and without (w/o) decompounding. In addition to the previously used peDOCS dataset, we apply keyphrase extraction approaches to the German datasets MedForum and Pythagoras. The final evaluation results will be influenced by two factors:

More accurate frequency counts As we have discussed before, the frequency counts will be more accurate, which should lead to higher quality keyphrases being extracted. This affects frequency-based rankings.

More keyphrase candidates The number of keyphrase candidates might increase, as it is possible that some of the parts created by the decompounding were not mentioned in the document before. This is the special case of a more accurate frequency count going up from 0 to 1.

We perform experiments to investigate the influence of both effects, first, the more accurate frequency counts, and second, the newly introduced keyphrase candidates.

More Accurate Frequency Counts

In order to isolate the effect, we limit the list of keyphrase candidates to those that are already present in the document without decompounding. We selected the MedForum dataset for this analysis, because a preliminary analysis has shown that it includes many compounds.

Table 3.9 shows improvements of evaluation results for five keyphrase extraction approaches on the MedForum datasets. The improvement is measured as the difference of evaluation metrics of using extraction approaches with decompounding compared to not using any decompounding. This table does not show absolute numbers, instead it shows the increase of performance. Absolute values are not comparable to other experimental settings, because all gold keyphrases that do not appear in the text as lemmas are disregarded. We can thus analyze the effect of more accurate frequency counts in isolation.

Results show that for tf-idf$_{\text{constant}}$, tf-idf, and tf-idf$_{web}$ our decompounding extension increases results. Decompounding does not affect results for the position baseline and TextRank as they are not based on frequency counting. For the frequency-based approaches, the effect is rather small in general, however consistent across all metrics and

Dataset	Decompounding w/o	w	Δ
peDOCS	.614	.632	.018
MedForum	.592	.631	.038
Pythagoras	.624	.625	.002

Table 3.10
Maximum recall for keyphrase extraction with and without decompounding for the datasets and all lemmas as candidates.

methods. We have thus shown that decompounding has indeed the potential to improve the performance of frequency-based keyphrase extraction approaches.

More keyphrase candidates

The second effect of decompounding is that new terms are introduced that cannot be found in the original document. Table 3.10 shows the maximum recall for lemmas with and without decompounding on all German datasets. Keyphrase extraction with decompounding increases the maximum recall on all datasets by up to 3.8% points. The increase is higher for the MedForum dataset while it is low for Pythagoras. Pythagoras comprises summaries of lesson transcripts for students in the ninth grade, thus teachers are less likely to use complex words which need to be decompounded. The smaller increase for peDOCS compared to MedForum is due to longer peDOCS documents. The longer a document is, the more likely a part in a compound also appears as an isolated token which limits the increase of maximum recall. peDOCS shows to have a higher maximum recall compared to collections with shorter documents because documents with more tokens also have more candidates. MedForum comprises forum data, which contains both medical terms and informal description of such terms. Furthermore, gold keyphrases were assigned to assist others in searching. This leads to having documents containing terms like *Augenschmerzen* (Engl.: *eye pain*) for which the gold keyphrase *Auge* (Engl.: *eye*) was assigned.

Combined results

Previously, we analyzed the effects of decompounding in isolation, now we analyze the combination of more accurate frequency counts and more keyphrase candidates on the overall results. Table 3.11 shows the complete results for the presented German datasets, described keyphrase extraction methods, and with and without the decompounding extension.

For the peDOCS dataset, we see a negative effect of decompounding. Only the position baseline and tf-idf$_{\text{constant}}$ benefit from decompounding in terms of mean average precision (MAP), while they yield lower results in terms of the other evaluation metrics. The improvement of the position baseline in terms of MAP might be to several correctly extracted keyphrases which have a lower rank. Precision and recall consider only the top-5 ranked extracted keyphrases, but R-precision considers the same number of extracted keyphrases as there are gold keyphrases. MAP, however, still considers lower ranked keyphrases (with lower influence) and can thus consider lower-ranked keyphrases. We

Dataset	Method	Decompounding											
		Precision@5			Recall@5			R-precision			MAP		
		w/o	w/	Δ	w/o	w/	Δ	w/o	w/	Δ	w/o	w/	Δ
peDOCS	*Upper bound*	.856	.864	.012	.393	.403	.010	.614	.632	.018	.614	.632	.018
	Position (baseline)	.096	.068	-.028	.042	.030	-.012	.092	.080	-.012	.083	.086	.003
	tf-idf$_{\text{constant}}$	.170	.160	-.010	.075	.070	-.004	.127	.125	-.002	.123	.123	.001
	tf-idf	.137	.117	-.020	.060	.051	-.009	.107	.088	-.019	.112	.099	-.014
	tf-idf$_{web}$	**.188**	.168	-.020	**.083**	.074	-.009	**.139**	.126	-.013	**.139**	.129	-.010
	TextRank	.153	.148	-.005	.067	.065	-.002	.112	.108	-.004	.116	.115	-.001
MedForum	*Upper bound*	.867	.890	.023	.397	.422	.025	.592	.631	.038	.592	.631	.038
	Position (baseline)	.082	.073	-.010	.049	.043	-.006	.101	.090	-.011	.142	.130	-.012
	tf-idf$_{\text{constant}}$	.149	.161	.012	.089	.096	.007	.144	.145	.001	.165	.162	-.003
	tf-idf	.235	**.282**	.047	.140	**.168**	.028	.210	**.234**	.025	.203	**.210**	.007
	tf-idf$_{web}$	.231	.165	-.067	.138	.098	-.040	.223	.159	-.064	.206	.180	-.027
	TextRank	.155	.171	.016	.092	.101	.009	.129	.141	.012	.149	.143	-.007
Pythagoras	*Upper bound*	.941	.942	.001	.344	.344	.001	.624	.625	.002	.624	.625	.002
	Position (baseline)	.030	.023	-.007	.014	.011	-.003	.044	.022	-.022	.106	.075	-.031
	tf-idf$_{\text{constant}}$	.137	.087	-.050	.066	.042	-.024	.143	.103	-.040	.153	.121	-.032
	tf-idf	.150	.150	.000	.072	.072	.000	.113	.114	.001	.141	.136	-.005
	tf-idf$_{web}$	**.187**	.100	-.087	**.090**	.048	-.042	**.205**	.102	-.103	**.191**	.136	-.055
	TextRank	.097	.060	-.037	.047	.029	-.018	.108	.069	-.039	.124	.095	-.029

Table 3.11
Complete results for keyphrase extraction approaches without (w/o) and with (w/) decompounding preprocessing.

have previously discussed that peDOCS has on average the longest documents and most likely contains all gold keyphrases multiple times in the document text. For this reason, frequency-based approaches do not benefit from additional frequency information obtained from compounds. On the contrary, more common keyphrases are weighted higher, which hurt results in the case of peDOCS with highly-specialized and longer keyphrases. As can be seen in Table 3.2, peDOCS has the keyphrases with most characters among the German datasets.

For the MedForum dataset, results improve with decompounding for tf-idf$_{\text{constant}}$ and tf-idf. As can be seen in Table 3.10, more accurate frequency counts improve results, and lead to a higher maximum recall (see Table 3.10). This also yields an improvement of results for the combination. Contrary to the other tf-idf configurations, results for tf-idf$_{web}$ decrease with decompounding. This leads to the observation that, besides the effect of more accurate ranking and more keyphrase candidates, a third effect influences results of keyphrase extraction methods: The ranking of additional keyphrase candidates obtained from decompounding. These candidates might appear infrequently in isolation and are ranked high if external document frequencies (df values) are used. Compound parts which do not appear in isolation[48]—hence, no good keyphrases—are ranked high in case of tf-idf$_{web}$ because their document frequency (df value) from the web is very low. In case of classic tf-idf they are ranked low because they are normalized with a frequency resource containing compound parts. In a preliminary step, the decompounding extension is applied to the entire document collection and used for normalizing term frequencies.

For the Pythagoras dataset, keyphrase extraction approaches yield similar results as for peDOCS. Decompounding decreases results, only results for tf-idf stay stable. As seen earlier (see Table 3.10), decompounding does not raise the maximum recall much (only by .002). As before in the case of the MedForum dataset, tf-idf$_{web}$ is influenced negatively by the decompounding extension. Results for tf-idf$_{web}$ decrease by .103 in terms of R-precision, which is a reduction of more than 50%. The ranking of keyphrases is hurt by many keyphrases, which appear as parts of compounds. They are ranked high because they infrequently appear as separate words.

Considering the characteristics of keyphrases in Pythagoras, we see that keyphrases are rather long with 12.22 characters per keyphrase. This is fewer than for the peDOCS dataset (12.27) but more than for the MedForum dataset (10.28). This leads to the observation that the style of the keyphrases has an effect on the applicability of decompounding. Datasets with more specific keyphrases are less likely to benefit from decompounding. For more general keyphrases, we see that decompounding improves the results of frequency-based approaches if their text frequencies are normalized with decompounded document frequencies.

Error analysis

To further gain insights into the effect of decompounding on keyphrase extraction, we analyze one example document from MedForum in detail. The document starts with the following;

> "Buchempfehlung: Stopp Diabetes
> Ernährungsempfehlung: Auf Menge und Qualität der KH achten, also die KH

[48]The verb *begießen* (Engl.: *to water*) can be split into the verb *gießen* (Engl.: *to pour*) and the prefix *be* which does not appear as an isolated word.

Method	Decompounding					
	Precision@5		Recall@5		R-precision	
	w/o	w/	w/o	w/	w/o	w/
Upper bound	$^2/_5$	$^3/_5$	$^2/_3$	$^3/_3$	$^2/_3$	$^3/_3$
Position (baseline)	$^1/_5$	$^2/_5$	$^1/_3$	$^2/_3$	$^1/_3$	$^0/_3$
tf-idf$_{\text{constant}}$	$^2/_5$	$^3/_5$	$^2/_3$	$^3/_3$	$^0/_3$	$^0/_3$
tf-idf	$^1/_5$	$^1/_5$	$^1/_3$	$^1/_3$	$^1/_3$	$^1/_3$
tf-idf$_{web}$	$^1/_5$	$^1/_5$	$^1/_3$	$^1/_3$	$^0/_3$	$^0/_3$

Table 3.12
Results for one example document from MedForum.

> moderat reduzieren und wenn dann bevorzugt auf Vollkornprodukte zurückgreifen..."

The document's gold keyphrases are *Diabetes* (Engl.: *diabetes*), *Bewegung* (Engl.: *movement*), and *Ernährung* (Engl.: *nutrition*). The document contains the two keyphrases as words and the keyphrase *Ernährung* as a part in the compounds *Ernährungsempfehlung* (Engl.: *nutrition advice*) and *Ernährungsumstellung* (Engl.: *nutrition change*).

The recall for the methods without decompounding is limited to $^2/_3$ because *Ernährung* cannot be extracted as a keyphrase candidate. With our decompounding processing, the recall can reach the maximum value of 1 because *Ernährung* has been split from the compounds.

Table 3.12 gives keyphrase extraction results for the mentioned example document in terms of precision, recall, and R-precision. Precision ranges from $^1/_5$ to $^3/_5$ as at maximum three of the 5 extracted keyphrases can be correct. tf-idf$_{\text{constant}}$ performs best in this example in terms of precision and recall. It achieves the maximum recall without and with decompounding processing. All correct keyphrases are in the top-5 most frequent terms in this document.

R-precision returns the precision of the top-3 extracted keyphrases (as there are three gold keyphrases). Position achieves a non-zero value only without decompounding, as the first three words of the document are *Buchempfehlung*, *Stopp*, and *Diabetes*, covering one of the gold keyphrases. When decompounding is used, the first word *Buchempfehlung* is split into the parts *Buch* and *Empfehlung*. Hence, *Diabetes* is no longer one of the first three keyphrase candidates in this document.

Most methods extracted *KH* (abbreviation for *Kohlenhydrat* (Engl.: *carbohydrate*) in the top-5 keyphrase candidates (it has the highest frequency and a high value for tf-idf), but was not assigned as a keyphrase by the annotators. tf-idf extracted the term *Vollkorn* (Engl.: *wholemeal*) as a keyphrase candidate and ranked it in the top-5 keyphrases. This part of the compound *Vollkornkost* was also extracted by one of the annotators and we believe that—like this—many of the high-ranked keyphrase candidates are potentially valuable keyphrases, but are not annotated as such.

3.8 Keyphrases in Text Structuring Scenarios

In our extensive analysis, we have seen that results of keyphrase identification approaches heavily vary depending on the dataset. We used different types of datasets in our experiments to analyze advantages of the approaches in specific scenarios. The datasets vary in respect to (i) the domain, (ii) the language, (iii) the length of the documents, and (iv) the style of the keyphrases and documents. We will discuss implications of these characteristics for every scenario we have defined in Section 2.3.

Focused searcher

A focused search is interested in answering questions on a specific topic and will most likely start with using a search engine. With the exception of the DUC-2001 dataset, all datasets cover a single domain, which means they are representative for this scenario. When accessing a closed domain document collection, a first clustering of documents based on their keyphrases (Gutwin, 1999) is helpful to get an overview of the topics. For this clustering, multi-label classification approaches to keyphrases are suitable because they assign a small set of keyphrases (only the most frequent ones) to all documents with high precision, which can then be used for clustering. However, this requires to already have manually assigned keyphrases for this collection or from a similar dataset. This is very often not the case. Without training data, tf-idf$_{web}$ performs well for both longer and shorter document collections. This is better suited because it does not need to count document frequencies in the entire collection, and frequent words in the collection are not filtered out. The normalization of tf-idf leads to many keyphrases being filtered out, because they appear throughout the entire collection and thus receive a high document frequency.

Another aspect is the language dependence. tf-idf$_{web}$ is language dependent, as it requires background frequencies. Common resources, e.g. Web1T, are available for several languages, but changing the resource requires additional processing components[49] and additional space for the data.

An extension for German, or other languages with many compounds, is using decompounding as preprocessing. However, this only improves results for document collections with shorter documents and a high number of compounds. With longer documents, the classic tf-idf configuration outperforms the decompounding extension.

Recreational news reader

We defined a recreational news reader as a user who starts his search at few entry news websites and typically reads further news from the same website.[50] This news website can be either in an open domain (e.g. world news such as BBC[51]), or closed domain (e.g. highly-specialized news for Apple products on MacRumors[52]). In open domain websites, generic keyphrases are required, like they are collected in the peDOCS dataset. peDOCS

[49] Some of the language processing components are already language dependent, but some language processing frameworks offer the possibility to exchange language-specific models during processing time (Eckart de Castilho and Gurevych, 2014).

[50] This phenomenon is described as stickiness by Li et al. (2006).

[51] `http://www.bbc.com/` (last accessed: 2014-12-07)

[52] `http://www.macrumors.com/` (last accessed: 2014-12-07)

contains generic keyphrases like *Deutschland* (Engl.: *Germany*), which allow for clustering the document collection into clusters. Accordingly, world news can be clustered into regional news based on the country keyphrases. For using a country keyphrases filter, we can either use a controlled vocabulary (as described in section 3.7.1) or apply multi-label classification based on country names as labels. Especially the second option is feasible for a larger news company which has the human resources to manually create the required training data.

For closed domain news, classic tf-idf is best suited because general keyphrases such as *Apple* or *Mac* are normalized by their document frequency in the document collection. With tf-idf$_{web}$, keyphrases are normalized with their external frequency, thus, leading to a much higher ranking of general keyphrases.

Knowledge worker in a company

A knowledge worker in a company is interested in specific information which is very often covered by a single or only few documents. The documents containing this information can usually be found in a corporate wiki or an Intranet. Finding these documents is a tedious tasks, because smaller wikis tend to be less structured (Désilets et al., 2005; Buffa, 2006). Thus, keyphrases need to be highly focused to the domain in which the company is active and potentially follow a taxonomy, or a controlled vocabulary.

Since the document collection is very often not as large as in the open domain (due to the restriction to a single company), a high recall for keyphrases is desired. In our experiments, the highest value for recall@5 was obtained with tf-idf with the decompounding extension on the MedForum dataset. It is possible to compute similarities to enhance the mapping of keyphrases to a search query (see Section 5.7).

3.9 Chapter Summary

The main contributions of this chapter can be summarized as follows:

Contribution 3.1: *We presented approaches to keyphrase extraction and keyphrase assignment and evaluated these with English and German datasets.*

Contribution 3.2: *We applied a preprocessing step for splitting compounds and evaluated keyphrase extraction approaches on a level including compounds.*

In this chapter, we analyzed keyphrase identification approaches for three German and three English datasets. We analyzed the peDOCS dataset in detail and compared it to the English datasets. peDOCS is the first dataset consisting of German documents with keyphrases that are manually assigned by professional indexers. Its number of documents and average number of keyphrases per document is larger than for other datasets.

We presented approaches to keyphrase assignment and analyzed their strengths and shortcomings. Keyphrase extraction approaches assign many potential keyphrases but are restricted to keyphrases that appear in the document. Multi-label classification assigns keyphrases with high precision but is limited to a predefined set of labels. Our experimental results on peDOCS showed that keyphrase extraction assigns keyphrases with low precision and high recall, while multi-label classification obtains higher precision and lower recall.

In addition, we presented a decompounding extension for keyphrase extraction. We created two further German datasets to analyze these effects and showed that decompounding increases results for keyphrase extraction on one German dataset. We identified two effects of decompounding relevant for keyphrase extraction: (i) changing the frequency counts of the compound parts, and (ii) potentially introducing terms into the document that were not mentioned initially. We find that the first effect slightly increases results when only updating the term frequencies. The second effect of decompounding is that new terms are introduced. Our results show that we are able to increase the maximum recall on all German datasets.

Results for the full approach (combining both effects) show an increase of up to .05 points in terms of precision@5. Besides the effects of more accurate ranking and more keyphrase candidates, the ranking of keyphrase candidates obtained from decompounding strongly influences results. Compound parts which do not appear in isolation are ranked high in case of tf-idf$_{web}$ because their document frequency (df-value) from the background corpus is very low. In case of tf-idf this is normalized with a frequency resource containing compound parts leading to an increase of results for keyphrase extraction.

For future work, we plan to verify our observations in a user study. Our hypothesis is that keyphrases identified by automatic approaches are comparable to manually assigned keyphrases and are useful for indexing and clustering of digital libraries, even if they are not part of the gold standard. We observed low inter-annotator agreement when creating a gold standard, showing that human ratings are not consistent. Additionally, Huang et al. (2009b) point out that human annotators perform on par with automatic approaches when evaluated with existing links in Wikipedia. Thus, humans are not a good model for this task as they (i) identify keyphrases subjectively, and (ii) two disjoint sets of keyphrases can still be good representations of the same document. For evaluating keyphrase identification in the future, we propose to use either pooling[53], or ask human annotators to rate possible keyphrases and weight identified keyphrases with these ratings.

Additionally, incorporating semantic resources such as UBY (Gurevych et al., 2012) may improve automatic evaluation for keyphrase identification by bridging the gap between related keyphrases. Considering a document has the gold keyphrase *Computer*, a keyphrase identification approach returning *PC* as keyphrase should receive a higher score than an approach returning *car*. Instead of using only correct and incorrect for evaluating identified keyphrases, a semantic evaluation may weight identified keyphrases using synonymy relations in WordNet (Fellbaum, 1998). Additionally, matches of keyphrases can be weighted using similarity measures (Bär et al., 2013).

We give a detailed description of the experimental framework used in our experiments in Section A.1 and we made the framework open-source to foster future research.[54]

[53] Pooling (Sparck Jones and van Rijsbergen, 1975) is an evaluation method used for information retrieval. First a list of keyphrases is collected from multiple approaches and then manually judged as relevant or irrelevant.

[54] `https://code.google.com/p/dkpro-keyphrases/` (last accessed: 2014-12-07)

Chapter 4

Text Structuring through Table-of-Contents

> *On two occasions I have been asked, — "Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?" In one case a member of the Upper, and in the other a member of the Lower, House put this question. I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question.* Charles Babbage in Passages from the Life of a Philosopher (Babbage, 1864, ch. 5 *Difference Engine No. 1*)

In this chapter, we deal with text structuring through a table-of-contents as a way to provide a reader with further information about the content and the structure of a document. Figure 4.1 shows a table-of-contents as one technique for text structuring.

4.1 Introduction

A table-of-contents (TOC) provides an easy way to gain an overview about a document as a TOC presents the document's content and structure. At the same time, a TOC captures the relative importance of document topics by arranging the topic titles in a hierarchical manner. Thus, TOCs might be used as a short document summary that provides more information about search results in a search engine. Figure 4.2 provides a sketch of such a search interface. Instead of a thumbnail[1] of the document like most search engines provide it, or a clustering of search results (Carpineto et al., 2009), we propose to use an automatically extracted TOC.

The task of automatically generating a table-of-contents can be decomposed in the subtasks of document segmentation, segment title generation, and hierarchy identification. The first step splits the document into topical parts, the second step generates an informative title for each segment, and the third step decides whether a segment is on a higher, equal, or lower level than the previous segment. This chapter presents novel approaches for the third subtask: hierarchy identification. Additionally, it presents a detailed analysis of results for segment title generation on the presented datasets.

Many documents are already segmented but only few documents already contain an explicit hierarchical TOC (e.g. Wikipedia articles), while for most documents it needs to

[1] A thumbnail is much smaller version of the original object, typically a picture or a website. Its size is reduced to fit literally on a thumbnail.

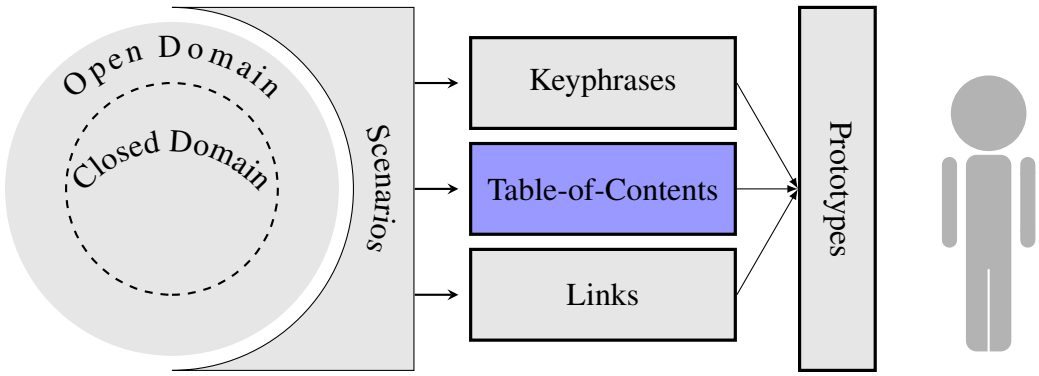


Figure 4.1: Graphical overview of the contents which are covered in this thesis with text structuring through table-of-contents highlighted blue.

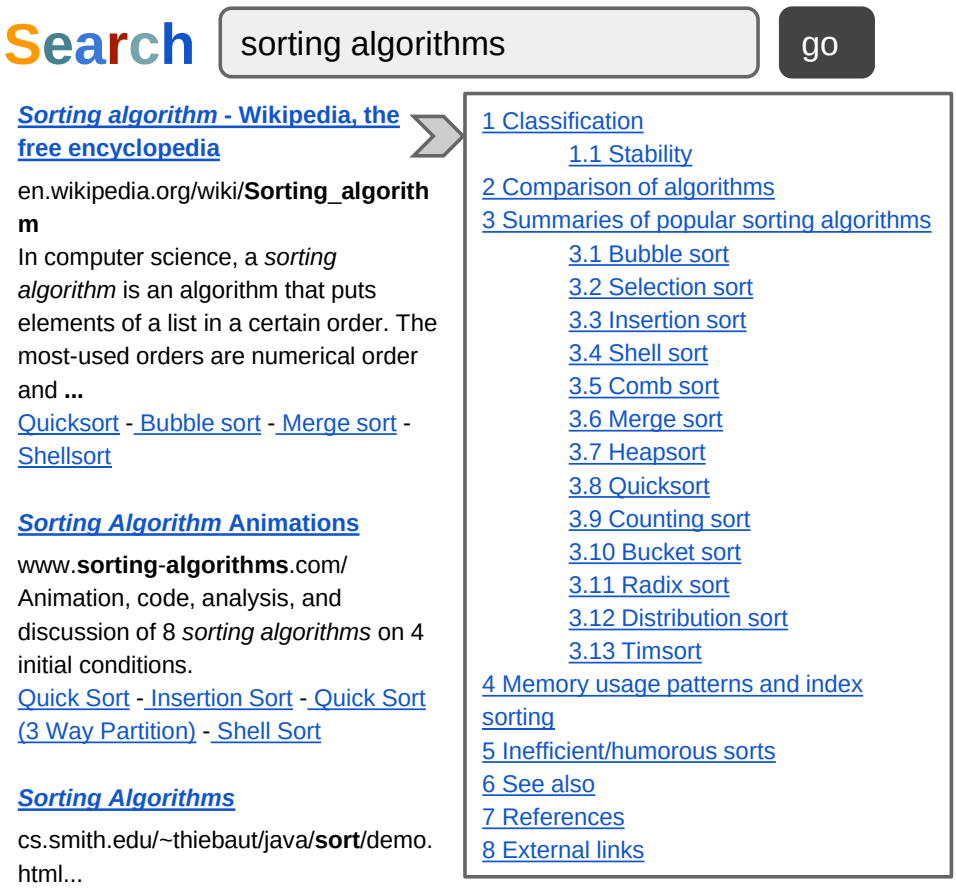


Figure 4.2: Mockup of a search user interface showing a table-of-contents along with the search results.

be automatically identified. For some documents, identification is straight-forward, e.g. if an HTML document already contains hierarchically structured headlines (`<h1>`, `<h2>`, etc). We focus on the most challenging case, in which only the textual content of the documents' segments is available and the hierarchy needs to be inferred using Natural Language Processing.

We automatically identify the hierarchy of two segments based on semantic and lexical

Figure 4.3: TOC of the chapter about table-of-contents generation.

features. We perform linguistic preprocessing including named entity recognition (Finkel et al., 2005), keyphrase extraction (Mihalcea and Tarau, 2004a) (see Chapter 3), and noun chunking (Schmid, 1994) which are then used as features for machine learning.

In this section, we develop new algorithms for segment hierarchy identification, present new evaluation datasets for all subtasks, and compare our newly developed methods with the state of the art. To foster future research, we present two new datasets and compare results on these datasets (see Section 4.3) with the one presented by Branavan et al. (2007) in prior work. We then compare the applicability of approaches with respect to the selected scenarios in Section 2.3.

4.2 Task Definition and Characteristics

Our system tackles the problem using a supervised classifier predicting the relation between the segments. Two segments can be on the *same*, *higher*, or *lower* level. Formally, the difference of a segment with level l_0 and a following segment with level l_1 is any integer n $\in [-\infty..\infty]$ for which $n = l_1 - l_0$. However, our analysis on the development data has shown that n typically is in the range of $[-2..2]$ which means that a following segment is at most 2 levels higher or lower than the previous segment.

Performance is measured in terms of accuracy and is defined as the ratio of correctly identified relations. A relation is correctly identified if the classifiers output n' ($n' \in [-\infty..\infty]$) equals the true level relation $n = l_1 - l_0$. For k segment pairs with a level relation of $l_{k+1} - l_k$, the accuracy is defined as

$$\text{acc.} = \frac{\sum_k \delta_{n,n'}}{k} \tag{4.1}$$

where $\delta_{n,n'}$ is the Kronecker delta function (returning one only if both indexes are equal, otherwise zero).

4.3 Resources

Branavan et al. (2007) extracted a single table-of-contents (TOC) from an algorithms textbook (Cormen et al., 2001) and split it into a training and a test set. We use the complete TOC as a test set and refer to it as *Cormen*. As a single TOC is a shallow basis for experimental results, we create two additional datasets containing tables-of-contents, allowing us to evaluate on different domains and styles of hierarchies.

We create the first dataset from a random selection of featured articles[2] in Wikipedia. They have been shown to be of high quality (Stein and Hess, 2007) and are complex enough to contain hierarchical TOCs. The following example is a shortened version of the XML representation for the Wikipedia article *solar eclipse*:[3]

<section id="1" title="Occurrence and cycles" level="1">
Total solar eclipses are rare events. Although they occur somewhere on Earth every 18 months on average, it has been estimated that they recur at any given place only once every 370 years, on average. [...]
<section id="2" title="Frequency per year" level="2">
Solar eclipses can occur 2 to 5 times per calendar year. Since the Gregorian calendar was begun in 1582, five solar eclipses occurred in 1693, 1758, 1823, 1805, 1870, 1935. The next occurrence will be 2206. [...]
</section>
<section id="3" title="Final totality" level="2">
Solar eclipses are an extreme rarity within the universe at large. They are seen on Earth because of a fortuitous combination of circumstances. Even on Earth, eclipses of the type familiar to people today are a temporary (on a geological time scale) phenomenon. [...]
</section>
</section>
<section id="4" title="Historical eclipses" level="1">
Historical eclipses are a valuable resource for historians, in that they allow a few historical events to be dated precisely, from which other dates and a society's calendar may be deduced. [...]
</section>

This excerpt contains four segments (sections) surrounded by XML section markup. The first segment (1^{st} level) is about the occurrence and cycles of solar eclipses and the following two subsegments (2^{nd} level) are about their frequency and totality. The last segment (1^{st} level) is again about the more general topic of historical eclipses. We can construct three pairs from this subset of the article. For the pair *1–2*, the level difference has the value 1, for pair *2–3* the value 0, and for pair *3–4*, it has the value -1.

We use a second dataset consisting of 55 books from the project Gutenberg.[4] We refer to these datasets as *Wikipedia* and *Gutenberg*. We annotated these datasets with the hierarchy level of each segment, ranging from 1 (top-level segment) to the lowest-level segment found in the datasets.

[2]Featured articles are specially flagged and considered as articles complying with the highest Wikipedia quality requirements (Stein and Hess, 2007).

[3]`https://en.wikipedia.org/wiki/Solar_eclipse` (last accessed: 2014-12-02)

[4]The same collection of books was used by Csomai and Mihalcea (2006) for experiments on back-of-the-book indexing. They mostly cover the domains humanities, science, and technology.

Dataset	doc	seg	$\varnothing \frac{tok}{seg}$
Cormen	1	607	733
Gutenberg	18	1,312	1927
Wikipedia	277	3,680	399

Table 4.1
Characteristics of evaluation datasets. Showing the total number of documents (doc), segments (seg) and average number of tokens in each segment ($\varnothing \frac{tok}{seg}$).

	Hierarchy level				
Dataset	1	2	3	4	5
Cormen	.00	.02	.08	.41	.48
Wikipedia	.07	.48	.41	.04	.00
Gutenberg	.01	.35	.49	.12	.03

Table 4.2
Distribution of segments over levels in the evaluation corpora.

Table 4.1 gives an overview of the datasets regarding the segment structure. Although the Cormen dataset consists of one book only, it is composed of more segments than an average document in any other dataset and thus is a valuable evaluation resource. The Wikipedia dataset contains on average the fewest tokens in each segment. It is the most fine-grained TOC. The Wikipedia and Gutenberg datasets cover a broad spectrum of topics, while the Cormen dataset is focused on computational algorithms.

4.3.1 Level distribution

Table 4.2 shows the distribution of levels in the datasets. The *Cormen* dataset has a much deeper structure compared to the other two datasets. The fraction of segments on the first level is below 1% because a single document may have only one top-level segment and this document contains far more than 100 segments. This is a special characteristic of this book: since it is often used to quickly look up specific topics, the authors provide a very fine-grained table-of-contents. In *Wikipedia*, most of the segments are on the second level. Articles in Wikipedia are rather short. According to the Wikipedia author guidelines a segment of a Wikipedia article is moved into an independent article if it gets too long. The *Gutenberg* dataset is more balanced as it contains documents from different authors. Similar to the Wikipedia dataset, most segments are on the second and third level.

Pair-wise level distribution

We now focus on the pairwise classification and investigate the pairwise relation of neighboring segments. Two segments on the same level have a hierarchy relation of $n = 0$, a segment that is one level higher than the following segment has a hierarchy relation of $n = 1$. Table 4.3 shows that for all datasets most of the segment pairs (neighboring segments) are on the same level. Although there are segments which are two levels higher

	Pairwise hierarchy relation				
Name	$n = 2$	$n = 1$	$n = 0$	$n = -1$	$n = -2$
Cormen	.00	.20	.60	.16	.03
Wikipedia	.00	.15	.71	.13	.01
Gutenberg	.00	.10	.80	.09	.01

Table 4.3
Distribution of pairwise level difference of segments of the evaluation corpora.

or three levels higher than the previous segment, this is the case for no more than 1% of all segment pairs. The Cormen dataset has the highest deviation of the level relation. This is due to the fact that its segments have a broad distribution of levels (see Table 4.2). Segments in the Gutenberg dataset, on the other hand, are in 80% of all cases on the same level as the previous segment. The case that the next segment is two levels lower, i.e. n=2, is very unlikely. This is in line with our expectations that a writer does not skip levels when starting a lower level segment.

4.4 Approaches to Table-of-Contents Generation

For some documents, the hierarchy of segments can be induced using HTML-based features. Pembe and Güngör (2010) focus on the DOM tree and on formatting features, but also use occurrences of manually crafted cue phrases such as *back to top*. However, most features are only applicable in very few cases where HTML markup directly provides a hierarchy. In order to provide a uniform user experience, a TOC also needs to be generated for documents where HTML-based methods fail or when only the textual content is available.

Feng et al. (2005) train a classifier to detect semantically coherent areas on a page. However, they make use of the existing HTML markup and return areas of the document instead of identifying hierarchical structures for segments. Besides markup and position features, they use features based on unigrams and bigrams for classifying a segment into one of 12 categories.[5]

For segment title generation, we divide related work into the following classes:

Unsupervised approaches make use of only the text in a segment. Therefore, titles are limited to words appearing in the text. They can be applied in all situations, but will often create trivial or even wrong titles.

Supervised approaches learn a model of document segments with their titles. They have a high precision, but require training data and are limited to an *a priori* determined set of titles for which the model is trained.

In the following, we organize the few available previous papers on this topic according to these two classes. The text-based approach by Lopez et al. (2011) uses a position heuristic. Each noun phrase in a segment is given a score depending on its position and its tf-idf value.

[5]These categories are mainly structure categories, e.g. forms, bulletined list, or Heading.

The supervised approach by Branavan et al. (2007) trains an incremental perceptron algorithm (Collins and Roark, 2004; Daumé and Marcu, 2005) to predict titles. It uses rules based on the hierarchical structure of the document[6] to re-rank the candidates towards the best global solution. Nguyen et al. (2009) expand the supervised approach by Branavan et al. (2007) using word clusters as additional features. Both approaches are trained and tested on the Cormen dataset. The book is split into a set of 39 independent documents at boundaries of segments of the second level. The newly created documents are randomly selected for training (80%) and testing (20%). Such an approach is not suited for our scenario of an end-to-end TOC creation, as we want to generate a TOC for a whole document and cannot train on parts of it. Besides, this tunes the system towards special characteristics of the book instead of having a domain-independent system.

Keyphrase extraction methods (Frank et al., 1999; Turney, 2000) may also be used for segment title generation if a reader prefers shorter headlines. These methods can be either unsupervised or supervised, as discussed in Chapter 3 of this thesis.

4.5 Experimental Setup

We create a supervised classifier for hierarchy classification, using different groups of features. We identified the following groups of features that solely make use of the text in each segment (we refer to these features as in-document features):

N-gram features We identify the top-500 most frequent n-grams[7] in the collection and use them as Boolean features for each segment. The feature value is set to `true` if the n-gram appears, `false` otherwise. These features reflect reoccurring cue phrases and generic terms for fixed segments like the introduction.

Length-based We compute the number of characters (including whitespaces) for both segments and use their difference as feature value. We apply the same procedure for the number of tokens and sentences. A higher-level segment might be shorter because it provides a summary of the following more detailed segments.

Entity-based We identify all named entities in each segment and return a Boolean feature if they share at least one entity. This feature is based on the assumption that two segments having the same entities are related. Two related segments are more likely on the same level or the second segment is a lower-level segment.

Noun chunk features Noun chunks in both segments are identified using the TreeTagger (Schmid, 1994), and then the average number of tokens for each of the segments is computed. The feature value is the difference of the average phrase length. Phrases in lower-level segments are longer because they are more detailed. In the example from Figure 4.2, the term *bubble sort algorithm* is longer than the frequently occurring upper level phrase *sorting algorithm*.

Additionally, the number of chunks that appear in both segments is divided by the number of chunks that appear in the second segment. If a term like *sorting algorithm* is the only shared term in both segments and the second segment contains

[6] An example of such a rule is the condition that neighboring segments must not have the same title.
[7] With n between 1 and 4.

in total ten phrases, then the noun chunk overlap is 10%. This feature is based on the assumption that lower-level segments mostly mention noun chunks that have already been introduced earlier.

Keyphrase-based We apply the state of the art keyphrase extraction approach TextRank (Mihalcea and Tarau, 2004a) and identify a ranked list of keyphrases in each segment. We compare the top-k ($k \in [1, 2, 3, 4, 5, 10, 20]$) keyphrases of each segment pair and return `true` if at least one keyphrase appears in both segments. These features also reflect topically related segments.

Frequency We apply another feature set which uses a background corpus in addition to the text of the segments. We use the Google Web1T corpus (Brants and Franz, 2006) to retrieve the frequency of a term. The average frequency of the top-k ($k \in [5, 10]$) keyphrases in a segment is calculated and the difference between two segments is the feature value. We expect lower-level segments to contain keyphrases that are less frequently used, because they are more specific.

We use WEKA (Hall et al., 2009) to train the classifier and report results obtained with SVM, which performed best on the development set.[8] We evaluate all approaches by computing the accuracy as the fraction of correctly identified hierarchy relations (see Section 4.2). As a baseline, we consider all segments to be on the same level. Even the baseline yields a high accuracy because most consequitive segments are on an equal level (majority class). This leads to flat documents without any hierarchies.

4.6 Experimental Results and Discussion

We evaluate the performance of our system using 10-fold cross-validation on previously unseen data using DKPro Lab (Eckart de Castilho and Gurevych, 2011) as the experimental framework for parameter sweeping. Evaluation results are shown in terms of accuracy as defined in Equation 4.1.

Table 4.4 shows our results on each dataset. Always predicting two segments to be on the same level is a strong baseline, as this is the case for 60% of cases in the Cormen and 80% of cases in the Gutenberg dataset. The table shows results for each of the feature groups defined in Section 4.5 numbered from (1) to (6). N-gram features perform best on the Cormen dataset while they perform worse than the baseline on the Wikipedia (WP) dataset. This difference might be due to the topic diversity in the Wikipedia and Cormen datasets. Wikipedia covers many topics, while Cormen is focused on a single topic (algorithms) and thus contains reappearing n-grams.

Noun chunk features are the best-performing group of features on the Wikipedia and Gutenberg and second best on the Cormen dataset. Entity, keyphrase, and frequency features do not improve the baseline in any of the presented datasets. Apparently, they are no good indicator for the hierarchical structure of document segments.

Combining all features further improves results on the Cormen dataset. However, the best results are obtained by combining all besides entity and keyphrase features. On the other two datasets (Wikipedia and Gutenberg), a combination of all features decreases

[8] We experimented with Naïve Bayes and J48, but the results were significantly lower.

	Cormen	WP	Gutenb.
Baseline (*always equal)*	.60	.71	.80
(1) N-gram features	**.86**†	.64	.86†
(2) Length features	.62	.76†	.80
(3) Entity features	.60	.71	.80
(4) Noun chunk features	.83†	**.86**†	**.91**†
(5) Keyphrase features	.60	.71	.80
(6) Frequency features	.60	.71	.80
All features	.86†	.77†	.86†
All features w/o (1)	.83†	**.86**†	**.91**†
All features w/o (3) & (5)	**.87**†	.77†	.86†

Table 4.4
Accuracy of approaches for hierarchy identification. Best results of feature groups and combinations are marked bold. Statistic significant improvements over the baseline are marked with a †. Statistical significance is computed using McNemar's test with $p \leq .001$. Upper bounds are not given because the gold standard was created by a single annotator. However, a pilot study revealed that the inter-annotator agreement reaches perfect agreement.

		Predicted				
		2	1	0	-1	-2
Actual	2	**0**	4	0	-	0
	1	0	**567**	0	0	0
	0	0	0	**2,585**	0	0
	-1	0	0	478	**0**	0
	-2	0	0	24	0	**0**

Table 4.5
Confusion matrix for best system (all features w/o n-gram features) on the Wikipedia dataset. Correctly identified segments are marked bold.

accuracy compared to a supervised system using only noun chunk features. The highest accuracy is obtained by using all features besides n-gram features.

Based on our observation that using all features performs worse than a selection of features, we analyzed the confusion matrix of the corresponding systems. Table 4.5 shows the confusion matrix for the best performing system from Table 4.4 on the Wikipedia dataset using selected features (all w/o n-gram features). The system is optimized towards accuracy and trained on unbalanced training data. This leads to a system returning either n$= 1$ (next level is one level lower) or n$= 0$ (same level). There are no cases where a lower-level segment is incorrectly classified as a higher-level segment but all cases with $|n| \geq 2$ are incorrectly classified as having a level difference of one.

Table 4.6 shows the confusion matrix for a system using all features on the same dataset as before (Wikipedia). The system also covers the case n$= -1$ (next level is one level higher), thus creating more realistic TOCs. In contrast to the previous system (see Table 4.5), some higher-level segment relations (n$<$0) are incorrectly classified as lower-level segment relations (n$>$0). Although the system using all features returns a lower

Actual \ Predicted	2	1	0	−1	−2
2	**0**	4	0	0	0
1	0	**539**	17	11	0
0	0	14	**2,115**	455	1
−1	0	1	323	**154**	0
−2	0	0	12	12	**0**

Table 4.6
Confusion matrix for a system using all features on the Wikipedia dataset. Correctly identified segments are marked bold.

1 Crew
1.1 Backup crew
1.2 Mission control
1.3 Mission insignia
2 Planning
3 Saturn V
4 Mission
4.1 Parameter summary
4.2 Launch and trans-lunar injection
4.3 Lunar trajectory
4.4 Lunar sphere of influence
4.5 Lunar orbit
4.5.1 Earthrise
4.6 Unplanned manual re-alignment
4.7 Cruise back to Earth and re-entry
5 Historical importance
6 Spacecraft location
7 In film

Correct TOC

1 Crew
1.1 Backup crew
2 Mission control
2.1 Mission insignia
3 Planning
4 Saturn V
5 Mission
5.1 Parameter summary
5.2 Launch and trans-lunar injection
5.3 Lunar trajectory
5.4 Lunar sphere of influence
5.5 Lunar orbit
5.5.1 Earthrise
5.5.2 Unplanned manual re-alignment
5.6 Cruise back to Earth and re-entry
6 Historical importance
7 Spacecraft location
8 In film

Predicted TOC

Figure 4.4: Correct and predicted TOCs of the article about Apollo 8 from the Wikipedia dataset (see `http://en.wikipedia.org/wiki/Apollo_8`, last accessed: 2014-12-07).

precision than the one using selected features, it better captures the way writers construct documents (also having segments on a higher level than previous segments).

Overall, results show that automatic hierarchy identification provides a TOC with a sufficient quality. To support this observation, Figure 4.4 shows the correct and predicted TOCs for the article about *Apollo 8* from the Wikipedia dataset. The correct TOC is on the left and the predicted TOC is on the right.

Section 1.3 (*Mission control*) was erroneously identified as being on a higher level

than the previous section. The system fails to identify that both segments are about the crew (backup and mission control crew). The section *Planning* is correctly identified as having a higher level than the previous segment but leading to a different numbering (5 instead of 4 due to earlier errors). Not all of the remaining segment relations are correctly identified but the overall TOC still provides a useful reference of the article's content. It allows a reader to quickly decide whether the article about *Apollo 8* fulfills her information need.

4.7 Segment Title Generation

So far, we have shown that our system is able to automatically predict a TOC for documents with segment boundaries and titles. In order to extend our system to documents that do not have titles for segments, we add a segment title generation step. News documents are very often segmented into smaller parts, but usually do not contain segment titles.[9]

We decided not to reuse existing datasets from summarization or keyphrase extraction tasks, as they are only focused on one possible style of titles (i.e. summaries or keyphrases). Instead, we apply our algorithms to the previously presented datasets for hierarchy identification (see Section 4.3) and analyze their characteristics in respect to their segment titles. The percentage of titles that actually appear in the corresponding segments is lowest for the Wikipedia dataset (18%) while it is highest for the Cormen dataset (27%). For the Gutenberg dataset, 23% of all titles appear in the text. The high value for the Cormen dataset is due to specific characteristics of textbooks that segment titles are repeated very often at the beginning of a segment.[10]

Frequency distribution of titles We further analyze the datasets in terms of segment counts for each title. Figure 4.5 shows the frequency of titles in the evaluation set on a logarithmic scale. We choose a random sample of 607 titles, which is the lowest number of titles in all three corpora, to allow a fair comparison across corpora. Hence, for the Cormen dataset, we used all 607 segment titles, for the other two (Wikipedia and Gutenberg) corpora we randomly selected 607 segment titles. For all three datasets, most titles are used for few segments.[11] For the datasets Wikipedia and Cormen, some titles are used more frequently. In comparison to that, the most frequent title of the Gutenberg dataset appears twice, only. Thus, we expect the supervised approaches to be most beneficial on the Wikipedia dataset. Due to the lack of training data, we cannot apply any supervised approaches on the Cormen dataset.[12]

4.7.1 Experimental Setup

Text-based approaches As simple baselines, we use the first token and the first noun phrase occurring in each segment. As a more sophisticated baseline, we rank tokens according to their tf-idf scores. Additionally, we use TextRank (Mihalcea and Tarau,

[9] For example, `cnn.com` (last accessed: 2014-12-07) uses *story paragraphs*.

[10] For example, the segment *Quicksort* begins with: *Quicksort is a sorting algorithm ...*

[11] Earlier, we made use of the most frequent keyphrases when applying multi-label classification to the task of keyphrase identification (see Section 3.7.1).

[12] Training and testing on 607 instances resulted in overfitting.

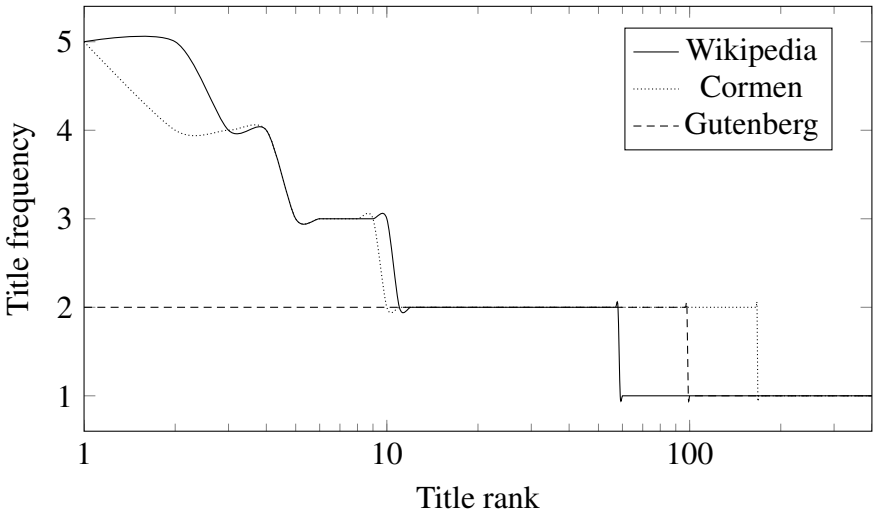


Figure 4.5: Frequency distribution of a random sample of 607 (the number of titles of the smallest dataset) segment titles on log-log-scale: it follows a power-law distribution.

2004a) to construct a graph representation with all noun phrases in the segment as nodes and weight edges according to their co-occurrence frequencies. After running PageRank (Page et al., 1999), the node with the highest score is selected as the segment title.[13]

As named entities from a segment are often used as titles, we extract them using the Stanford named entity tagger (Finkel et al., 2005) and take the first one as the segment title.[14]

Supervised approaches We train a text classification model, which is based on character 6-grams[15] as features. In Wikipedia, most articles have sections like *See also*, *References*, or *External links*, while books usually start with a chapter *Preface*. We restrict the list of segment title candidates to those appearing at least twice in the training data.

In contrast to previous approaches (Branavan et al., 2007; Nguyen et al., 2009; Jin and Hauptmann, 2001), we do not train on parts of the same document for which we want to predict titles, but rather on full documents of the same type (Wikipedia articles and books). This is an important difference, as in our usage scenario we need to generate full TOCs for previously unseen documents. On the Cormen dataset, we cannot perform a training phase, since it consists of a single book.

Evaluation Metrics We evaluated all approaches using two evaluation metrics. We propose **accuracy** as the evaluation metric. A generated title is counted as correct only if it exactly matches the correct title. Hence, methods that generate long titles by adding many important phrases are penalized.

The **Rouge** evaluation metric is commonly used for evaluating summarization systems. It is based on n-gram overlap, where—in our case—the generated title is compared to the gold title. We use Rouge-L which is based on the longest common subsequence.

[13] See Section 3.4 for further explanations on keyphrase extraction approaches.

[14] We also experimented using the most frequent entity but achieved lower results.

[15] A previous evaluation has shown that 6-grams yield the best results for this task on all development sets. We used LingPipe (`http://alias-i.com/lingpipe`, last accessed: 2014-12-07) for classification.

Approach	Type	Wikipedia Acc.	Wikipedia Rouge-L	Gutenberg Acc.	Gutenberg Rouge-L	Cormen Acc.	Cormen Rouge-L
Branavan et al. (2007)	*Supervised*	-	-	-	-	-	*.249*
Nguyen et al. (2009)		-	-	-	-	-	*.281*
Position (token)	Baselines	.007	.034	.004	.078	.010	.137
Position (NP)		.012	.112	.037	**.180**	**.061**	**.364**
tf-idf		.017	.057	**.042**	.094	.020	.206
TextRank	Unsupervised	.014	.058	.011	.060	.012	.195
Named entity		.006	.046	.011	.065	.000	.037
Text classification	Supervised	**.133**	**.169**	.004	.008	*	*

Table 4.7
Results for segment title generation. No results for supervised text classification on the Cormen dataset are shown since no training data is available. The set of segment titles for Cormen is largely disjoint from the set of segment titles from the other datasets, thus, supervised approaches cannot assign any titles.

This metric is frequently used in previous work for evaluating supervised approaches to generating TOCs because it considers near misses. We believe that it is not well suited for evaluating title generation, however, we use it for the sake of comparison with related work.

4.7.2 Experiments and Results

Table 4.7 shows the results of title generation approaches on the three datasets. On the Cormen dataset, we compare our approaches with two state of the art methods. For the newly created datasets, no previous results are available.

Using the first noun phrase returns the best titles on the Cormen dataset, which is in agreement with our observation from Section 4.7.1 that many segments repeat their title in the beginning. This also explains the high performance of the state of the art approaches which are also taking the position and part-of-speech of candidates into account. Branavan et al. (2007) report about a feature for the supervised systems eliminating generic phrases without giving examples of these phrases.

Supervised text classification approach works quite well in case of the Wikipedia dataset with its frequently repeated titles. The approach does not work well on the Gutenberg dataset, as segments such as *Preface* treat different topics in most Gutenberg books. Consequently, the text classifier is not able to learn the specific properties of that segment. In future work, it will be necessary to adapt the classifier in order to focus on features that better grasp the function of a segment inside a document. For example, the introduction of a scientific paper always reads "introduction-like", while the covered topic changes from paper to paper. This is in line with research concerning topic bias (Mikros and Argiri, 2007; Brooke and Hirst, 2011) in which topic-independent features are applied.

The overall level of performance in terms of accuracy and Rouge seems rather low. However, accuracy is only a rough estimate of the real performance, as many good titles might not be represented in the gold standard and Rouge is higher when comparing longer texts. Besides, a user might be interested in a specialized table-of-contents, such as the one consisting only of named entities. For example, in a document about US presidential

elections, a TOC consisting only of the names of presidents might be more informative than the one consisting of the dates of the four-year periods. A flexible system for generating segment titles enables the user to decide about which titles are more interesting and thus increases the user's experience.

4.8 Table-of-Contents in Text Structuring Scenarios

We used three very distinct datasets in our analysis of table-of-contents. The first dataset is an open domain dataset obtained from randomly selected featured articles in Wikipedia. The documents follow a rather fixed structure. The Gutenberg dataset is also an open domain dataset and comprises complete books. The structure heavily varies depending on the type[16] and the style of writing.

Focused searcher

The focused searcher first tries to get an overview of a topic by browsing through several documents. Hereby, a table-of-contents is a great improvement because it provides the opportunity to quickly look into one document and then move to the next one. Not only the topic, but also the structure is revealed. The Cormen dataset is a good example of a closed domain dataset with educational content. A student in preparation for an exam about algorithms can use the hierarchies given by a table-of-contents to structure his/her learning efforts.

For an automatic generation of such a table-of-contents, the full supervised classification for hierarchy identification can be applied, which yields an accuracy of 86%. Using the first noun phrase in a segment as its title resulted in a Rouge-L score of .364, which is also the best result across all datasets.

Recreational news reader

A recreational news reader does not need to structure a single document because there is no specific information especially interesting. However, helping a new reader to decide what to read is a useful application. Instead of having a flat-structured news website, a user may cluster the news reports based on their category (e.g. sports or economy) and create a hierarchical representation from the most general to the most specific ones. So far we have not investigated this possibility. To the best of our knowledge, there exists no related work on the hierarchical structuring of multiple documents.[17]

Knowledge worker in a company

A knowledge worker looks for one specific piece of information, and any hint pointing her/him to this information is a support. The hierarchical overview from a table-of-contents can provide hints where to look for this information. Like in the Wikipedia

[16] All books are non-fiction, but cover different topics of science and technology, e.g. botany.

[17] The closest work we found is about multi-document summarization to create an overview page of a topic for Wikipedia (Sauper and Barzilay, 2009).

dataset, a corporate wiki or Intranet has a limited set of segment titles. We discussed before (see Section 3.8) that a supervised classification system yields good results for identifying keyphrases in a corporate setting. The same applies to segment titles. A supervised system allows for assigning a segment title from a manually created list of possible titles. This helps to structure company information further, as segments of many documents can be combined to get an overview across many documents.

4.9 Chapter Summary

The main contributions of this chapter can be summarized as follows:

Contribution 4.1: *We presented and evaluated approaches for identifying hierarchy relations between segments for a table-of-contents.*

Contribution 4.2: *We investigated characteristics of segment titles in table-of-contents and analyzed the performance of existing keyphrase identification approaches for automatically identifying segment titles in a table-of-contents.*

We presented the first study on automatically identifying the hierarchical structure of a table-of-contents for different kinds of text (articles and books from different domains). The task of *segment hierarchy identification* is a new task which has not been investigated for non-HTML text. We created two new evaluation datasets for this task, used a supervised approach based on textual features and a background corpus, and significantly improved results over a strong baseline.

For documents with missing segment titles, *generating segment titles* is an interesting use case for keyphrase extraction and text classification techniques. We applied approaches from both tasks to the existing and two new evaluation datasets. We showed that the performance of approaches is still quite low. Overall, we have shown that for most documents a TOC can be generated by detecting the hierarchical relations if the documents already contain segments with corresponding titles. In other cases, one can use segment title generation, but additional research based on our newly created datasets will be necessary to further improve the task performance.

In future work, we plan to develop an interactive system prototype with the user interface and perform user tests. Furthermore, we plan to continue developing better features for the task of hierarchy identification, and creating methods for post-processing a TOC in order to generate a coherent table-of-contents.

We made the newly created evaluation datasets and our experimental framework publicly available in order to foster future research on the table-of-contents generation.[18]

[18] Available at
`http://www.ukp.tu-darmstadt.de/data/table-of-contents-generation/hierarchy-identification/` (last accessed: 2014-12-07)

Chapter 5

Text Structuring through Links

> *The real problem in speech is not precise language. The problem is clear language. The desire is to have the idea clearly communicated to the other person. It is only necessary to be precise when there is some doubt as to the meaning of a phrase, and then the precision should be put in the place where the doubt exists. It is really quite impossible to say anything with absolute precision, unless that thing is so abstracted from the real world as to not represent any real thing.*
>
> *Pure mathematics is just such an abstraction from the real world, and pure mathematics does have a special precise language for dealing with its own special and technical subjects. But this precise language is not precise in any sense if you deal with real objects of the world, and it is only pedantic and quite confusing to use it unless there are some special subtleties which have to be carefully distinguished.*
>
> Richard Feynman

In this section, we focus on text structuring through links. Links create connections between documents, thus providing the means to look up further information for terms in linked documents. Being able to look up further information results in clear language, as ambiguities can be resolved. Figure 5.1 shows links as one technique for text structuring.

5.1 Introduction

The Internet is based on many web pages that are connected by links. Without links, it would be impossible to quickly navigate from one page to another. Additionally, search algorithms like HITS (Kleinberg, 1999) or PageRank (Page et al., 1999) utilize links to determine the relevance of pages, making it easier to find redundant information.

Some regions of the Web already contain a large number of links. For example, links in Wikipedia are created by a large community of highly motivated contributors (Nakayama et al., 2007). In other situations, however, (e.g. in corporate Intranets or wikis) it is more difficult to motivate people to make contributions (Majchrzak et al., 2006). Users find it especially difficult to add links, as they need to decide what other pages constitute a valid link target. Even in smaller document collections like a corporate Intranet, this is a very difficult task, especially if the pages are subject to frequent

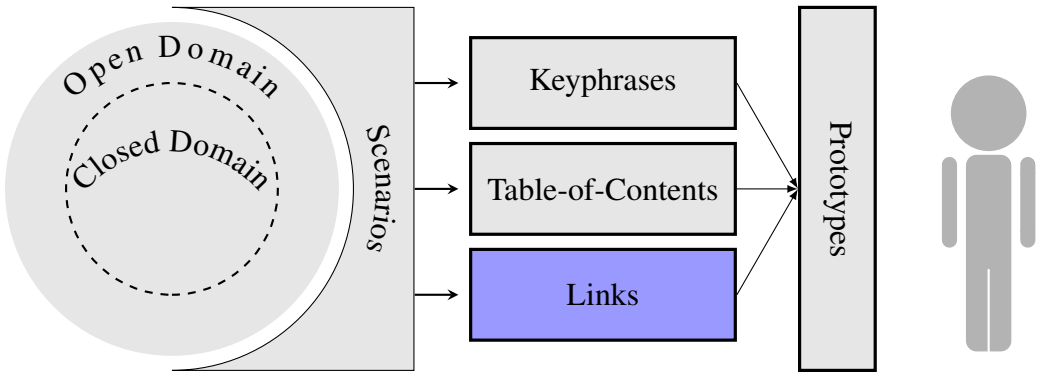


Figure 5.1: Graphical overview of the contents which are covered in this thesis with text structuring through links highlighted blue.

changes. In such a situation, approaches for automatically identifying links provide support for users trying to add new links to a certain document collection. Thereby, a link identification algorithm usually first selects promising link anchors in a document, and then retrieves possible target documents for every anchor. The user only has to decide whether the suggested link should be added to the document.

Current state of the art link identification approaches can be categorized according to the type of prior knowledge they utilize, e.g. already existing links, meaningful page titles, or the document text. In this chapter, we argue that previous evaluations of link identification approaches have always used document collections like Wikipedia in which a large amount of prior knowledge in form of links and meaningful page titles is available. This obviously entails a bias towards approaches using the available information which consequently outperform approaches using no prior information, by a wide margin. However, in many realistic settings (e.g. in corporate Intranets) one cannot rely on already existing links or page titles (Buffa, 2006). We show that the common evaluation setting which uses Wikipedia is a rather special case.

In this chapter, we focus on one special case of link identification: Linking entities, also known as *named entity disambiguation*. This task imposes further challenges, including the selection of a suitable sense inventory. The purpose of link identification as presented in this chapter, is to resolve ambiguity in words, thus, making language clearer for both humans and computer. Resolving ambiguity is the process of deciding in which sense a word is used, thus transforming words to senses (see Section 2.6.2). Ambiguity may reside in a word having any part-of-speech. According to WordNet version 3.1, the word *bright* has eleven word senses, including a person being smart and a bright sun. Studies have shown that even for human annotators the distinction between rather fine-grained senses is very hard (Véronis, 1998).

There is a large body of work on word sense disambiguation (Agirre and Edmonds, 2006), however, we focus on disambiguating named entities. It is the process of linking named entities to a sense inventory, often referred to as entity linking if Wikipedia is the target sense inventory. Specifically, it is the task of linking a mention in a text (e.g. *Washington*) to the correctly disambiguated representation of the entity in some knowledge base (e.g. either *George Washington* or *Washington, D.C.* depending on what *Washington* was referring to in this context). In this chapter, we focus on named entity linking due to (i) its clear cut senses, and (ii) high information content of entities for finding information.

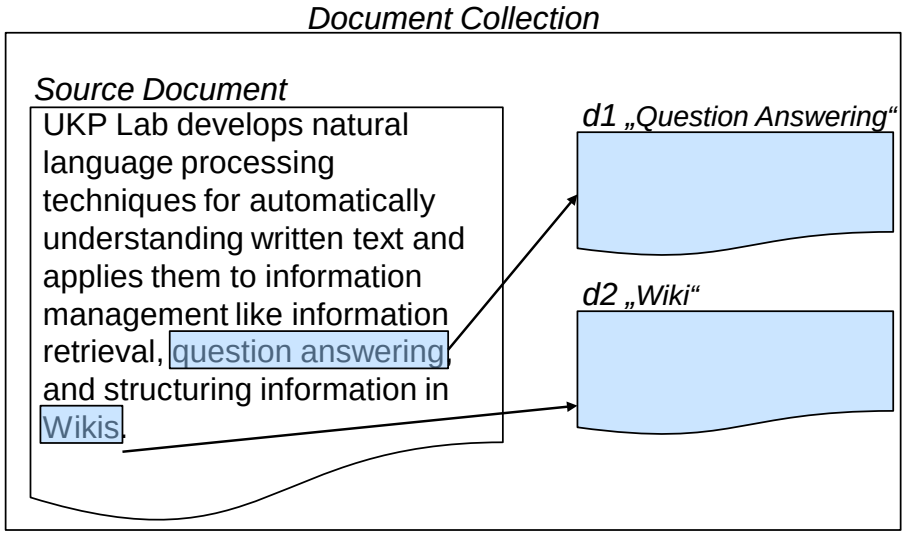


Figure 5.2: Linking from text in a source document to target documents.

Clear cut senses Entities are easier to distinguish than different senses of a verb, an adjective, or an abstract noun. Véronis (1998) has shown that the inter-annotator agreement for assigning senses is actually higher for nouns, than for verbs or for adjectives (moderate κ agreement of .46 for nouns as opposed to .41 for verbs and adjectives).

High information content In many application scenarios, especially in those scenarios described in Section 2.3, people are interested in finding information about entities, e.g. the city of Paris, or the actor George Clooney. It is actually the nouns, especially the named entities, which define the topic of a document. Barr et al. (2008) found that 40% of all search queries are proper nouns and over 70% of the search queries analyzed are nouns.[1]

5.2 Task Definition and Characteristics

In this section, we will use the following notation in formulas and texts: Each text document d in a collection D contains i mentions (or surface strings or anchors) m_i with a context c_i. For each mention m_i, there exists a set of entities (or targets or senses) E_i, which are entity candidates for m_i. Consider the link in Figure 5.2 that connects the anchor *question answering* with the target document d_1 that represents e_1.

We define a link as a mention in a source document which is connected to another entity. We then define $l(m, e)$ as the number of links where m is a mention in a source document and $e \in E$ is the entity (e.g. a document) the mention is pointing to.

Depending on the task investigated, different terms are used. In the domain of word sense disambiguation (Agirre and Edmonds, 2006), senses are identified for words. For the task of link discovery, e.g. in the context of the *INEX Link the Wiki track* (Huang et al., 2007, 2008, 2009a; Trotman et al., 2010), anchors are linked to targets. Finally, in the domain of named entity disambiguation, mentions of named entities are linked to

[1]For keyphrase extraction, it has been shown that noun phrases, either the full phrase (Wang and Li, 2010) or their heads (Barker and Cornacchia, 2000), are the best keyphrases.

entities. These entities need to be included in a sense inventory. In the remainder of this chapter, we use the term mentions to refer to the strings which should be linked and entities to refer to the targets for the links.

The standard evaluation measure for disambiguation tasks is accuracy, defined as the number of correct links compared to the total number of links (Agirre and Edmonds, 2006; McNamee and Dang, 2009). For the task of identifying mentions for links, we will use precision, recall and F-measure.[2] As opposed to evaluating the disambiguation of mentions, we compare a list of automatically identified mentions with a list of manually identified mentions (the gold standard). We thus rely on these evaluation measures from information retrieval, which have previously shown to be adequate in the context of evaluating keyphrase extraction (see Chapter 3).

5.3 Resources

In this section, we present resources for link identification. We first define and present sense inventories and then describe the commonly used sense inventory Wikipedia. Later, we describe the TAC-KBP dataset in detail, which comprises an entity linking dataset with Wikipedia as the sense inventory.

5.3.1 Sense Inventories

A sense inventory is a list of entities (or senses) representing meanings. Every human has a unique sense inventory in mind. It depends on the cultural background, experience, and education. A human's sense inventory is constantly changing, mainly increasing the coverage as new entities emerge, but also forgetting infrequently used senses. For computers, there exist several machine-readable sense inventories.[3] These sense inventories include WordNet (Fellbaum, 1998), GermaNet (Hamp and Feldweg, 1997), VerbNet (Kipper et al., 2008), FrameNet (Ruppenhofer et al., 2010), BabelNet (Navigli and Ponzetto, 2012), and online community-produced resources such as Wikipedia[4] and Wiktionary (Meyer and Gurevych, 2012b). They differ in the way they are constructed and in relation to their purposes. Wikipedia and Wiktionary are collaboratively constructed inventories with descriptions of their entries, which means that basically everyone can edit information and add senses. WordNet, VerbNet, FrameNet, and BabelNet are expert built sense inventories which provide additional information. WordNet (English) and GermaNet (German) provide semantic relations between their synsets, VerbNet organizes English verbs according to roles, FrameNet contains semantic frames, and BabelNet is a network of concepts and semantic relations. Giles (2005) found that the quality of content is actually comparable in collaboratively constructed resources and expert-build ones.[5] This has been confirmed in later work (Gurevych and Wolf, 2010).

[2] The F-measure is defined as $F_1 = \frac{2PR}{P+R}$.

[3] Not all of them were constructed to be machine-readable.

[4] `https://www.wikipedia.org/` (last accessed: 2014-12-07)

[5] In the study, he compares the Encyclopedia Britannica (`http://www.britannica.com/`, last accessed: 2014-12-07) with Wikipedia.

[6] English version from 3rd of July, 2008

[7] Only pages are counted which are marked as articles.

Sense inventory	senses	nouns		verbs		adjectives		adverbs	
WordNet 3.0	117,659	82,115	70%	13,767	12%	18,156	15%	3,621	3%
GermaNet 8.0	84,584	64,380	76%	11,024	13%	9,180	11%	0	0%
VerbNet 2.0	5,000	0	0%	5,000	100%	0	0%	0	0%
Wikipedia[6]	2,183,497[7]	-	-	-	-	-	-	-	-

Table 5.1
Number of senses and the distribution of part-of-speech tags for each sense inventory.

One of the main characteristics of a given sense inventory is the distribution of words with different part-of-speech tags. Table 5.1 shows the number of entries and their part-of-speech tags. WordNet and GermaNet have a high number of entries (called synsets) and a high proportion of nouns. Wiktionary, on the other hand, is organized differently. Entries are organized as articles, which may contain multiple senses with a different part-of-speech. This results in fewer entries and the sum of the distribution is above 100% as one article may have more than one part-of-speech. VerbNet contains verbs based on the classification by Levin (1993). Wikipedia is organized similar to Wiktionary with articles as entries. It contains by far the most entries and is focused on nouns, but it does not encode their part-of-speech information in the article.

Wikipedia is best suited for the task of entity linking, since it contains the largest number of entries from all presented inventories, is organized with one article for every sense, and contains the highest ratio of nouns, of which many are named entities.

Structure of Wikipedia

Wikipedia is a collaboratively constructed resource, covering entities and events of public interest. Due to its collaborative nature, there exist various none-article pages for the purpose of discussion and organization. Among those are pages to discuss the content in one article, redirects from one term to another article, disambiguation pages, and category pages. For all pages, there also exists a history of revisions, which has been subject to research. Ferschke et al. (2013) analyzed the collaborative writing process, and Daxenberger and Gurevych (2013) analyzed edit-turn pairs in the Wikipedia revision history.

For the task of entity linking, the following pages are of special interest: articles, redirect pages, and disambiguation pages. Articles are the senses of Wikipedia in our sense inventory. Redirects provide alternative names for these senses, e.g. the page *Obama* redirects to *Barack Obama*. Finally, disambiguation pages show a list of possible senses for a mention and very often give a short summary of the senses' definition.

Accessing Wikipedia

Wikipedia provides online access via several interfaces. However, as Eckart de Castilho (2014) points out, web interfaces provide neither reproducibility, nor reliability. We thus access Wikipedia by using the dumps provided by the Wikimedia Foundation[7] and use JWPL (Zesch et al., 2008a) to retrieve its contents. JWPL allows for filtering any kind of pages, e.g. retrieving only the senses. It also offers the possibility to extract the article text without any wiki markup for using it as the sense description.

[7] `http://wikimediafoundation.org/` (last accessed: 2014-12-07)

Corpus	Type	Size			
		Person	Organization	GPE	Sum
TAC-KBP 2009	Newswire	627	2,710	567	3,904
TAC-KBP 2010	Web data (train)	500	500	500	1,500
	Newswire (test)	500	500	500	1,500
	Web data (test)	250	250	250	750

Table 5.2
Size of TAC-KBP datasets in terms of number of mentions per entity type.

5.3.2 TAC-KBP datasets

In this section, we describe the TAC-KBP datasets, which we will use in our experiment for entity linking. The TAC-KBP datasets are a special case of link identification datasets, as they contain only named entities for linking.

The TAC-KBP stands for the *Knowledge Base Population* (KBP) task which is held annually at the *Text Analysis Conference* (TAC). The Knowledge Base Population is a shared task at the Text Analysis Conference (McNamee and Dang, 2009; Ji et al., 2010, 2011) has released evaluation datasets since 2009. The datasets comprise a set of documents, where each document has a document ID, an entity string which occurs at least once in that document. Additionally, there is a set of queries defined, which contain an id, the ID of the document it refers to, and the mention (name) and position of the entity in the document. The following example is about the mention Robinson:

```
<query id="EL_000304">
 <name>Robinson</name>
 <docid>APW_ENG_20080416.1425.LDC2009T13</docid>
 <beg>565</beg>
 <end>572</end>
</query>
```

The gold standard consists of a list with all query IDs and the corresponding entity IDs from the knowledge base, which is a subset of the 2008 dump of Wikipedia. Due to the annual TAC-KBP workshop there exist several datasets, which are used for evaluating the participating systems' performances.

Table 5.2 shows an overview of the TAC-KBP datasets from 2009 and 2010. The TAC-KBP 2010 dataset is divided into several smaller datasets depending on the type of the source documents. The web data was further divided into a training and test set. Participants of the TAC-KBP workshops could rely on previous datasets as additional training data. Each document in the TAC-KBP datasets contains exactly one named entity, which should be linked to the knowledge base. This is different to related tasks such as the SemEval all words word sense disambiguation task (Agirre et al., 2010), in which all words in a document should be disambiguated. With a single entity in a document already marked, it is possible to evaluate the performance of entity linking with the TAC-KBP datasets in isolation from the entity recognition task.

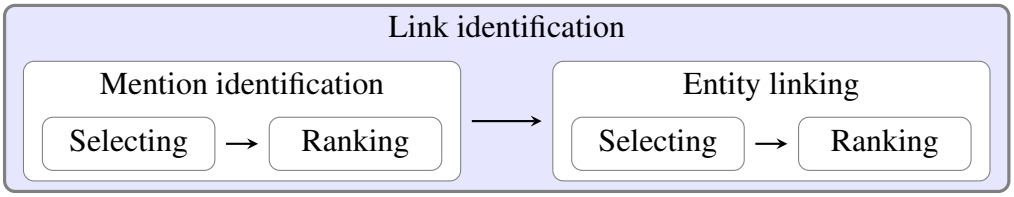


Figure 5.3: Overview of link identification with a division into mention identification and entity linking.

5.4 Approaches to Link Identification

We described that the task of link identification actually consists of two subtasks: (i) identifying mentions, and (ii) linking of entities for the identified mentions. Each of these subtasks can again be divided into a selection and a ranking step. Figure 5.3 gives an overview of link identification. There are approaches covering both subtasks, or dealing only with entity linking. This section is organized into these subtasks and steps, giving an overview of the corresponding approaches.

We will further divide the approaches into the categories *text-based*, *title-based*, and *link-based*. In the default case (which is also the most difficult one), approaches can only make use of raw text in a document (text-based) and no other prior knowledge. In some document collections (including Intranets and wikis), each document has a title that can be used to facilitate the anchor selection process (title-based). If the collection contains links between documents, this is another important source of information (link-based). The latter can be seen as a supervised approach because it requires links as training data.

5.4.1 Mention identification

Mention identification extracts text spans that can be used as a link anchor by first selecting a set of candidates and then ranking them. The methods used for this purpose are very similar to *Keyphrase Extraction* (see Chapter 3). A notable difference is that keyphrases should describe the topics in the document, while good mentions provide a starting point for a link to another related document. For example in a document about *Baseball*, good keyphrases are *Pitcher* and *Home run*, while *famous players* is not a good keyphrase. This mention, however, is a good anchor for linking to a document with a list of famous baseball players.

Selecting mentions

In this step, a list of candidate mentions (i.e. phrases from the document) is selected. We categorize the approaches according to the amount of prior knowledge used.

Text-based A widely used approach to mention selection is to select all possible candidate mentions that consist of a certain number of tokens (called *n-grams*) (Manning and Schütze, 1999). As this approach creates a lot of invalid anchors (e.g. *is the yellow* is a valid n-gram with very low probability of being a valid anchor), it heavily relies on the subsequent *mention ranking* step to filter such cases. Linguistic preprocessing components like noun phrase chunking (Schmid, 1994) or named entity tagging (Finkel et al.,

2005) can be used to restrict the anchor candidates to a subset that is more likely to contain valid mentions. For example, if we restrict anchors to noun phrases, *is the yellow* would have not been selected as a mention candidate, because it is not a valid noun phrase, while e.g. *yellow submarine* is valid and has a higher probability of being a useful mention. A special case of noun phrases are named entities that correspond to predefined categories like persons (e.g. *George Washington*), locations (e.g. *New York*), or organizations (e.g. *United Nations*). Using only named entities further filters the list of selected anchors, as it also rejects common noun phrases like *the beginning*. A major disadvantage of linguistically motivated mention selection approaches is that noun phrase chunking or named entity tagging are not available for all languages and need to be trained for the specific document collection.

Title-based If the documents in a collection contain titles, they can be used to constrain the list of selected mention candidates. This has two advantages. First, titles are usually well formed phrases. Thus, the list of mention candidates can be pruned without the need to apply linguistic preprocessing tools that might not be available for all languages. Second, in the subsequent *entity linking* step, there will always be a document whose title exactly matches the mention, and that is thus a very likely entity for this mention. The downside is that titles are not available for all document collections, which limits the applicability of this approach. Also, it does not cover cases in which mentions are highly related to page titles, e.g. synonyms. For these cases, we need to apply approaches using sense similarity (see Section 5.7).

Link-based If a document collection already contains links, the set of corresponding link anchors constitute a good source of mention candidates. A phrase that has already been selected by a human as an anchor in one document is probably still a good mention in another document. This also solves the problem that good mentions are sometimes unusual phrases. However, it also means that, in order to reliably add links, the document collection already needs to contain links which turns the task into a "chicken or egg" dilemma.

Ranking mentions

The output of the mention selection step is a (possibly noisy) list of candidates that needs to be ranked in order to select the best candidates. Taking the full list of mention candidates might result in an over-linked article. However, the optimal number of links per document depends on the user preferences[8] and the domain.[9] Thus, we need to rank the full list of anchor candidates in order to return only a certain number of top-ranked anchors that are necessary in that context. Like the anchor selection approaches, we categorize anchor ranking approaches according to the amount of prior knowledge that is used:

[8]For keyphrases, which are very similar to link anchors, it has been shown that the favored density of keywords depends on the user (Tucker and Whittaker, 2009).

[9]In Wikipedia for example, on average 6% of all words are used in links (Mihalcea and Csomai, 2007).

Text-based There are three commonly-used text-based approaches for mention ranking: the length of the anchor phrase, the tf-idf score (see Section 3.4.1) of the mention, and using TextRank (Mihalcea and Tarau, 2004a).[10]

Length: The length of a mention candidate can be used as a baseline ranking method. Longer candidates correspond to longer page titles which are assumed to be more specific than others (and thus better mentions) (Geva, 2007). For example, the longer candidate *Queen of England* is more specific than *England*, and should be ranked higher. The usefulness of this approach strongly depends on the mention selection strategy. Obviously, it will not work well, if mention candidates are all of equal length (e.g. if only single tokens are selected), as all candidates will have the same rank. This applies to using n-grams as keyphrases as well.

tf-idf: The underlying hypothesis of this approach is that more frequently appearing candidates are more likely to be good mentions (see Section 3.4.1). However, in order to avoid ranking common words too high, frequency should be combined with the inverse document frequency (df), which is high if a candidate only appears in few documents. For example, if the terms *the* and *United Nations* both appear five times in a document, both would be ranked the same using only the text frequency. However, *the* probably occurs in almost every document. Hence, its document frequency (df value) will be very high, while *United Nations* only appears in a couple of documents resulting in a lower document frequency. Overall, *United Nations* will be ranked much higher than *the* due to the higher combined tf-idf score.

TextRank: This method by Mihalcea and Tarau (2004a) creates a graph representation of a document. In the graph, anchor candidates are used as the nodes, and an edge is added if two candidates co-occur in a certain context window (e.g. 3 words left or right of the anchor candidate, or in the same sentence as the anchor candidate) in the document. The weight of the edge is defined as the number of co-occurrences. A graph centrality measure like PageRank (Page et al., 1999) is then used to rank the anchor nodes. TextRank has shown to provide reliable results for the task of keyphrase identification (see Section 3.7).

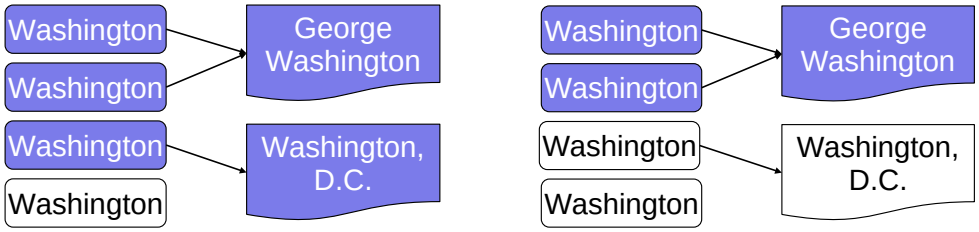
Title-based So far, no special methods relying on the page title knowledge have been proposed for *mention ranking*.

Link-based The methods in this section make use of the links that are already present in the document collection. As described in Section 5.2, a link is a mention m pointing to an entity e and $l(m, e)$ is the number of links from m to e.

Mention probability estimates the probability of a mention m in a document to be used in a link based on how often it has been previously used as a mention for any entity ($\Sigma_e l(m, e)$) divided by the total number of documents in which the phrase appeared ($|D_m|$). D_m is defined as the subset of all documents containing the mention m.

$$\text{mention probability}(m) = \frac{\Sigma_e \, l(m, e)}{|E_m|} \tag{5.1}$$

[10] For mention ranking, we rely on the approaches from keyphrase extraction as described in Chapter 3.



(a) Anchor probability (b) Anchor strength

Figure 5.4: Example of anchor phrases partially used as link anchors for multiple target documents.

Considering the example shown in Figure 5.4(a), the phrase *Washington* is used three times in a link and occurs four times in the text. Thus, its mention probability is $\frac{3}{4}$. Mention probability is also called *keyphraseness* in the domain of keyphrase extraction (Mihalcea and Csomai, 2007). Milne and Witten (2008b) use it as an important feature for their machine learning approach to *mention ranking*.

Mention strength (Itakura and Clarke, 2007) A disadvantage of *mention probability* is that it does not consider whether a mention is ambiguous. For example, if a mention is used in a link in each document, its *mention probability* will be 1.0 even if it points to a different entity every time. A common example is the mention *here* in sentences like "The documents can be found here." where the link might point to almost any entity. Thus, *mention strength* also incorporates the ambiguity of a mention by only counting the number of times a link points to its most frequent entity.

$$\text{mention strength}(m) = max_{e \in E} \frac{l(m,e)}{|D_m|} \tag{5.2}$$

For example in Figure 5.4(b), the mention *Washington* points twice to the entity e_1 *George Washington* and only once to the entity e_2 *Washington, D.C.* As the anchor is still used four times in the collection, the resulting *mention strength* is $^2/_4$ or $^1/_2$.

5.4.2 Entity linking

Entity linking identifies the best matching entity for a mention. As for mention identification, the task consists of a selection and a ranking step. This task is similar to information retrieval, where relevant entities are retrieved given a query (in this case the mention). The first step is candidate generation in which for every mention a set of possible entities is created. These lists of entities are then filtered and ranked. Figure 5.5 gives a list of different approaches for entity linking. Due to the Knowledge Base Population[11] (KBP) track at the Text Analysis Conference (TAC), there exists a diversity of approaches for both entity selection and entity ranking.

[11] `http://www.nist.gov/tac/2014/KBP/index.html` (last accessed: 2014-12-07)

Selecting entities

For every mention m, there exists a set of possible entries E in a sense inventory. The best matching entity from this set should be used for linking. A naïve approach is to select the entire set of senses in the sense inventory, thus resulting in optimum recall but poor precision.

Text-based Following information retrieval approaches, the target documents are indexed using common search engine libraries like Terrier (Ounis et al., 2006) or Lucene (McCandless et al., 2010). The resulting index is then queried using the mention. However, with this approach only exact matches of the mention in the sense inventory can be found. More relevant target documents could be retrieved using semantic search (Gurevych et al., 2007b), where highly related terms like synonyms are taken into account. Another approach is to use query expansion, e.g. by using more context (like e.g. the first sentence from the source document) (Sunercan and Birturk, 2010). A third approach is to compute semantic similarity between the mention and the entity (see Section 5.7).

Title-based Instead of searching the full document text, we can constrain the search space to document titles only. Especially for huge collections, this results in a faster response time which is of great importance in an online system for link identification. However, only searching in the space of document titles also means that there needs to be an overlap between the mention and the document title.

Link-based If the document collection already contains links, we can check if there are already links with a given mention. We can then limit the list of entity candidates to those that have already been linked from this mention. For example, if the mention *this approach* occurs in the collection pointing to the entities *Dijkstra Algorithm* and *Breadth-First-Search*, only these two entities will be considered as entity candidates. Han and Zhao (2009) first create a dictionary based on Wikipedia using redirects, disambiguation pages, and link anchors. This kind of dictionary approach is followed by Chang et al. (2010) and Spitkovsky and Chang (2012).

Chang et al. (2010) collect a Wikipedia-based dictionary that lists all mentions with a frequency distribution of their entities. It is generated by adding all categories, redirects, and all links in all Wikipedia articles of one language version. Statistical data slightly differs depending on the time stamp of the Wikipedia version. Probability may change over time since Wikipedia articles are under constant revision, e.g. they are added, split, merged, or removed. Hence, different Wikipedia versions may return different frequency distributions.

Spitkovsky and Chang (2012) collect information from Google search logs and therefore are able to provide statistics from a large crowd of Internet users. It also includes information about Wikipedia inter-language links, and types (e.g. disambiguation pages) of the entity page in Wikipedia.

As extension to those approaches, acronym expansion detects mentions like *MJ* for Michael Jackson (Zheng et al., 2010). Further approaches are using *Did you mean?* in Wikipedia (Zhang et al., 2010), or GIZA++ (Och and Ney, 2003) for entity linking (Han and Sun, 2011).

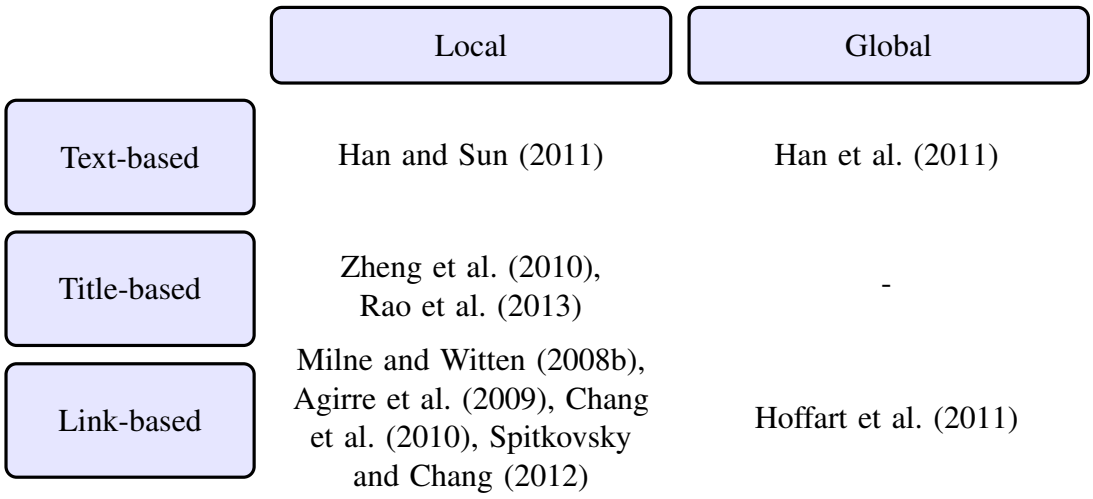


Figure 5.5: Overview of related work based on disambiguation level (local and global) and required resources (mention, sense description, link information).

Ranking entities

We organize approaches for ranking the entity candidates as a matrix based on whether entities are disambiguated on a local or global level. Local disambiguation considers only one entity at a time, while global disambiguation finds the best disambiguation for all entities at the same time. Additionally, we divide approaches based on the resources they require for disambiguating an entity. Figure 5.5 shows an overview of related work organized in this matrix. Local approaches consider one mention at a time, hence a single document contains only a single mention m. Global approaches use the context of the mention and disambiguate the marked mention and all further mentions in the context collectively, seeking to optimize the relatedness between the entities. For each mention in the context and the marked mention itself, every possible entity candidate is mapped to a Wikipedia article. The context is transformed from a list of words to lists of Wikipedia articles, then typically a graph-based algorithm is used to find optimal links.

Text-based If a search engine is used for entity selection, the resulting list is usually the full sense inventory with assigned relevance scores. Typically, a search engine indexes the article text for the entities and returns a ranked list of entities based on the mention as the query.

Finding the correct entity boils down to word sense disambiguation (Navigli, 2009). For example, the anchor *Bank* may match the document titles *Bank (river)* and *Bank (money)*.[12] We now have to determine which article is meant by measuring the similarity between the textual content in which the anchor phrase is used and the textual content of the document.

Text-based approaches can make use of the entity's and mention's contexts. A common approach in word sense disambiguation is to use the context of the mention to identify the correct entity. For instance, this is done by computing similarity with the Lesk

[12]Such situations often occur in Wikipedia, as there is a high probability that more than one article exists for a certain term. For example, there are 8 different articles for *Einstein* in the English Wikipedia version of September 8th, 2010.

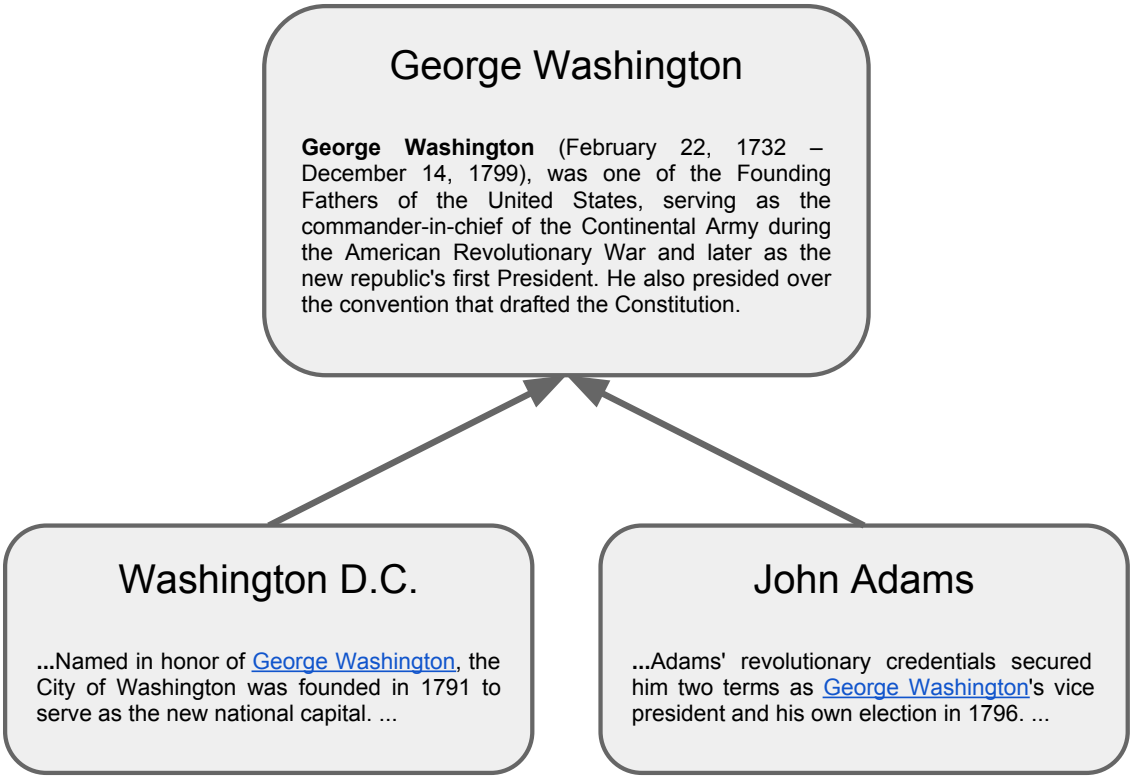


Figure 5.6: Constructing a context for the entity *George Washington*, using text from the Wikipedia article test and using the contexts of Wikipedia links, which refer to that entity.

algorithm (Lesk, 1986). The words in the context of the mention are compared to words in the description of each entity given by its Wikipedia article, e.g. by computing the cosine of a bag-of-word representation (Mihalcea and Csomai, 2007; Medelyan et al., 2008). Han and Sun (2011) weight context words using the frequency in the description, smoothed by their n-gram frequency (Jelinek and Mercer, 1980), and compare them to definitions of entity candidates. Using smoothing leads to high-frequent words having a lower influence than uncommon words, similar to tf-idf weighting (see Section 3.4.1).

As an alternative, one can use the context of the mention of a link pointing to a Wikipedia article, instead of using the description in the corresponding Wikipedia article (Han and Sun, 2011; Erbs et al., 2012). Figure 5.6 shows an example for *George Washington* which is—among many others—linked from the articles *Washington D.C.* and *John Adams*. Either the document text of the article can be used or the context of the link anchors pointing to the article for *George Washington*. Pedersen et al. (2005) proposed this approach for clustering. A window of 50 words to either side of the mention is used to define the context of the mention.

A related approach is done by Monahan et al. (2011) who use the Google Normalized Distance (Cilibrasi and Vitanyi, 2007) as a similarity metric to compare link mentions with the local context. Google Normalized Distance could potentially also be used to rank entity candidates based on their name.

Title-based Title-based approaches rank entity candidates by the string representation of the mention and the entity. A similarity-based approach computes the score $\text{sim}(m, e)$

according to the string similarity between the mention and the title of the candidate entity (Zheng et al., 2010). This approach is especially useful in case the mention is misspelled, e.g. the wrongly written name *George Waschington* is much closer to *George Washington* than to *Washington D.C.* Rao et al. (2013) apply the popularity of the entity title as search term for filtering irrelevant entities. Entities get discarded if their Wikipedia articles do not appear in the top-20 Google results.

Link-based One way to incorporate link-based information into entity ranking is to count how often the selected mention points to that entity (*entity strength*). Formally, we define:

$$\text{entity strength}(m, e) = \frac{l(m, e)}{\Sigma_{e' \in E}\, l(m, e')} \tag{5.3}$$

Consider the mention *Washington* in Figure 5.4 with the possible target documents d_1 *George Washington* and d_2 *Washington, D.C.*: The target strength for *George Washington* is $\frac{2}{3}$ and for *Washington, D.C.* it is $\frac{1}{3}$. We thus use the frequency information in the link-based dictionaries. We use frequency information from the dictionary by Chang et al. (2010), which contains information from Wikipedia links. An alternative dictionary (Spitkovsky and Chang, 2012) also contains information obtained from Google query logs showing which Wikipedia article has been clicked for which query.

Bunescu (2006) presented the earliest work on entity linking using Wikipedia. They choose the entity according to the similarity between the context of the mention and the text of the articles, using tf-idf and cosine similarity. In addition they enriched the term vector of each article with words from articles in the same category. Varma et al. (2009) also use a local algorithm. They use the occurrences anchors in Wikipedia to train a supervised classifier per target string according to the context of occurrence of the mention, and the correct hyperlink as provided by the anchor.

There is a large body of work dealing with global optimization for link-based approaches (Gabrilovich and Markovitch, 2007; Strube and Ponzetto, 2006; Milne, 2007; Milne and Witten, 2008b; Hoffart et al., 2009; Sunercan and Birturk, 2010; Shen et al., 2012). Milne and Witten (2008b) use a global algorithm, which compares how closely related two articles are based on the number of incoming links shared between the articles.[13] Cucerzan (2007) uses context vectors consisting of key words and short phrases extracted from Wikipedia. This system attempts to disambiguate all named entities in a single context simultaneously, adding the constraint, that all target Wikipedia articles should be from the same Wikipedia category. Hachey et al. (2011) combine subgraphs of the article hyperlink graph with the similarity between the context of the target mention and the article text. Unfortunately they do not report results for the subgraph method on its own.

Kulkarni et al. (2009) propose a method which collectively resolves the wikification task in a document as an optimization problem, using ideas from Milne and Witten (2008b) and Cucerzan (2007). They disambiguate by applying hill-climbing algorithms and integer linear programming. They thus use information which includes the text of the articles, hyperlinks, and the category system.

Lehmann et al. (2010) use a supervised system, which combines features based on mention and candidate name similarity, as well as context similarity. The information

[13] They reuse the methods from (Milne and Witten, 2008a).

they use includes hyperlinks, categories, context similarity and relations from info boxes.

Hoffart et al. (2011) use a hybrid system, which integrates entity-entity coherence, entity priors, mention-entity similarity and robustness. The authors try several methods for computing each of the measures but only report results for some of them. The system uses supervised machine learning on a large number of withheld development documents.

Erbs et al. (2012) apply the Personalized PageRank algorithm (Agirre and Soroa, 2009) to the task of entity linking. They use Wikipedia articles as nodes in the graph and links as edges.[14] This is a link-based approach, based on the PageRank (Page et al., 1999) algorithm. The PageRank random walk algorithm is a method for ranking nodes in a graph based on their relative structural importance. First, PageRank constructs a graph representation $G = (N, E)$, taking entities as nodes N and relations or links as edges E. Second, a random walker traverses the graph, but at any time jumps to another node with a *teleport probability*. The difference between standard PageRank and Personalized PageRank is the construction of the *teleport probability*. In standard PageRank, it is a uniform probability, i.e. every node receives the same probability. In Personalized PageRank, every node has an individual teleport probability, specified by a *teleport vector*. For entity linking, the *teleport vector* can be given by the target strength (see Equation 5.3). The final weight of node $n \in N$ represents the proportion of time the random particle spends visiting it after a sufficiently long time, and corresponds to that node's structural importance in the graph. Because the resulting vector is the stationary distribution of a Markov chain, it is unique for a particular walk formulation. As the teleport vector is non-uniform, the stationary distribution will be biased towards specific parts of the graph. The final weight of a node is the score of an entity, and the one with the highest score is selected as the target for a mention.

Supervised Systems at TAC-KBP

The Knowledge Base Population workshop of the Text Analysis Conference provides a framework for comparing different approaches to entity linking in a controlled setting. Most of the participating teams in the last years have shifted to using supervised approaches. The increase of quality of the resulting entity linking systems is partially due to the increasing amount of training data available from the previous years. There is a wide variety of supervised approaches, e.g. Zheng et al. (2010) apply ListNet (Cao et al., 2006) and Ranking Perceptron (Shen and Joshi, 2005) for a supervised ranking of entity candidates.

5.4.3 Classifying existing approaches to link identification

In this section, we give an overview of state of the art link identification approaches and classify these according to the information they utilize.

Gevas Page Name Matching (GPNM)

GPNM (Geva, 2007) combines methods which are text, title, and link-based. Every title from the document collection that can be found in the source document is considered a mention candidate that is then ranked according to its length. Entities are selected and

[14] Agirre and Soroa (2009) use synsets and relations from WordNet (Fellbaum, 1998).

ranked using a link-based approach, i.e. possible entities are limited to those that have already been linked using this mention. The score is set according to how often they have been linked using this mention.

Itakura & Clark Link Mining (ICLM)

This approach (Itakura and Clarke, 2007) completely relies on knowledge derived from existing links. In the source document, all phrases that have at least once been used in the document collection in a link are considered as mention candidates. The candidates are ranked using *mention strength*. Entity candidates are all documents that have been previously linked using this mention. The more frequent an entity has been linked by the mention, the better its rank is in the list of entities.

5.5 Experimental Setup

We evaluate approaches in two experimental settings. First, we evaluate link identification in a Wikipedia setting. The Wikipedia link structure is taken as gold standard and approaches should be evaluated as to whether they can reconstruct Wikipedia links. Second, we evaluate entity linking approaches for named entities. In this setting, a single named entity for every document is marked, which means that no mention identification needs to be performed. The approaches are evaluated in respect with their performance for entity linking.

5.5.1 Identifying internal Wikipedia links

Automated evaluation of link identification approaches requires a document collection that already contains links. Previously, Wikipedia has been used for that purpose, e.g. for evaluation in the context of the INEX 2009 Link-the-Wiki-Track (Huang et al., 2009a). We follow this approach, as Wikipedia is publicly available and contains high quality links that were collaboratively added and verified by a large number of Wikipedia contributors.[15]

We use a Wikipedia snapshot from October 8th, 2008 containing 2,666,190 articles and 135,478,255 links (this is the same version that was used for the evaluation at INEX 2009 (Huang et al., 2009a)). The dataset used for testing consists of 2,709 articles (every 1,000th article) containing 140,143 gold standard links. The remaining articles are used to provide the link and title knowledge.

Mention identification

For mention identification, we propose a text-based approach without using any prior knowledge about the document collection. Hence, we do not make use of any document titles or existing links. We experiment with three mention selection methods (tokens, n-grams, and noun phrases) and three entity ranking methods (mention length, tf-idf, and TextRank).

[15] As the Wikipedia community follows specific guidelines when adding links (see `http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking`, last accessed: 2014-12-07), we can consider Wikipedia as a corpus that is annotated with high quality.

We evaluate the performance of *mention identification* and *entity linking* separately. In order to allow a fair comparison of the different mention identification approaches, we evaluate them using the same preprocessing and postprocessing steps. We rely on the software repository DKPro Core (Eckart de Castilho and Gurevych, 2014), especially components from Stanford Core NLP (Manning et al., 2014) (e.g. segmentation, tagging, and lemmatization (Toutanova and Manning, 2000; Toutanova and Klein, 2003)).

We limit the number of suggested anchor candidates per document, as some anchor selection methods like *n-gram* create too many candidates. We use a threshold relative to the document length instead of a fixed threshold, as the document length varies considerably in the test collection. For example, adding only ten links to a long document might not be sufficient, while adding ten links to a short document might already be too much. Following (Mihalcea and Csomai, 2007), we use a threshold of 6% (i.e. approximately 1 out of 17 tokens is used as a mention for a link) as an upper bound. In our Wikipedia evaluation dataset, however, we found that the average number of tokens used as links is only 1.7%. Thus, we decided to also use a threshold of 1% (i.e. 1 out of 100 tokens is used as a link anchor) as a lower bound.

We use two baselines: (i) selecting all tokens as anchor candidates, and (ii) selecting all noun phrases. In both cases, we rank the candidates according to their position in the document (the earlier a candidate appears, the better it is ranked). We compare the baselines with the state of the art approaches GPNM (title-based) and ICLM (link-based), as well as a wide range of text-based configurations as explained in the previous section.

We evaluated results in terms of precision[16], recall[17], and f-measure[18] at the two linking threshold levels 1% and 6%.

Entity linking

For entity selection, we perform a full text search in the document collection with the mention text as the query. We use the target relevance scores returned by the search engine Terrier (Ounis et al., 2006) for entity ranking.

Entity linking cannot be evaluated fully independently of mention identification, as we need a list of mentions for which to discover the correct targets. Thus, we select the best performing mention identification configuration for each approach from the 6% threshold case, and perform entity linking using the mention candidates output by using the same type of knowledge.[19] Using the best set of mentions for every approach allows for a fair comparison. As taking into account entity linking for wrong mention candidates would yield misleading results, we only consider correct mentions. We also filter mentions that are dates or numbers. This way, we mimic our setting of first selecting a mention, and then choosing the best matching entity, where a user would only select valid mentions from the full list of suggested mentions.

As evaluation metric for entity linking, we use a relaxed version of accuracy: We compute a result set with entity suggestions which is defined to be correct if it contains the gold entity. This relaxed definition mimics the user's view on the result set, as we

[16] # correct anchors retrieved / # total anchors retrieved

[17] # correct anchors retrieved / # anchors in gold set

[18] $F = \frac{2PR}{P+R}$

[19] We also tested mention candidates produced by other approaches, but it did only marginally influence results.

expect the user to identify the correct entity given a list of suggestions. Obviously, this is limited to a result set up to a certain length. Hence, we limit the result set to ten entity suggestions, as this is the number of suggestions returned by common search engines. The overall accuracy is then calculated as the number of target sets containing the correct entity divided by the total number of gold entities.

5.5.2 Linking to a knowledge base with Personalized PageRank

For the second evaluation setting, we rely on the TAC-KBP datasets from 2009 and 2010. The 2009 and 2010 datasets have been used most in the entity linking literature. We used the 2009 dataset for development and 2010 for testing, as done in previous other works (Hachey et al., 2012). The knowledge base for the TAC-KBP is taken from the English Wikipedia version from April, 5th 2008 (KB timestamp).

As before, preprocessing components are taken from the open source project DKPro Core (Eckart de Castilho and Gurevych, 2014). We use components for tagging, lemmatization (Toutanova and Manning, 2000; Toutanova and Klein, 2003) and named entity recognition (Finkel et al., 2005) from Stanford Core NLP (Manning et al., 2014).

For evaluation we use accuracy, the ratio between correctly disambiguated mentions and total number of entities with a link to the Knowledge Base in the dataset. This corresponds to *non-NIL accuracy* at TAC-KBP and *Micro precision@1.0* in (Hoffart et al., 2011). Note that this evaluation ignores cases where the mention has no respective entity in the inventory (so-called NIL cases), as that is out of the scope of our entity linking approach.

We define an upper bound and a baseline based on the dictionary obtained from the respective Wikipedia version. The upper bound is defined by the maximum possible accuracy with a perfect ranking. The dictionary does not contain the correct entity for all mentions; for some mentions, no entities are returned or the correct entity is not included in the list of entity candidates. The baseline is defined as the most frequent sense returned by the candidate. The most frequent sense baseline is a strong baseline for entity linking as it yielded good results in previous TAC-KBP tasks (McNamee and Dang, 2009; Ji et al., 2010, 2011). We use the Wikipedia knowledge base from March, 12th 2008 and apply the Personalized PageRank algorithm for entity linking.

Entity ranking

In Section 5.4, we presented approaches for entity linking. We propose to use Personalized PageRank for weighting entities in a graph. We start from a Wikipedia dump, retaining article pages and discarding redirect, disambiguation and category pages, and mining hyperlinks between articles. As a result, we end up with $3,635,342$ articles and $68,012,306$ hyperlinks between them. In order to link running text to the articles in the graph, we also need a dictionary, i.e., an association between string mentions with all possible articles the mention can refer to. The dictionary is a key component for generating the candidate articles for a mention. Ideally, a dictionary provides a balance between precision and recall to capture the correct article for a mention while maintaining a small set of candidates. It can be constructed by adding the title of the article, the redirect pages, the disambiguation pages, and the anchor texts from Wikipedia links.

Anchors are indeed a rich source of information (Hoffart et al., 2011). For instance, links to the page *Monk* are created by using textual anchors such as *lama* or *brothers*. As a result, the dictionary entries for those mentions will contain the *Monk* article. Some authors include additional information for building the dictionary, such as hat-notes (the most frequent referent on a disambiguation page), the terms of the article's first paragraph (Hachey et al., 2012), or even the anchor texts linking into Wikipedia from non-Wikipedia Web pages (Lehmann et al., 2010; Chang et al., 2010; Spitkovsky and Chang, 2012). Yazdani and Popescu-Belis (2013) include titles, redirects and anchor texts for creating the dictionary. When a string mention does not match the dictionary, they also mapped it to an article using string similarity techniques.

We use the dictionary from Chang et al. (2010) based on the same Wikipedia dump, using article titles, redirects, disambiguation pages, and anchor text. Mentions are lowercased and all text between parenthesis is removed. If the mention links to a disambiguation page, it is associated with all possible articles the disambiguation page points to. Each association between a mention and an article is scored with the prior probability, estimated as the number of times that the mention occurs in the anchor text of an article divided by the total number of occurrences of the mention (see Equation 5.3). The dictionary disambiguates a mention by returning the entity with the highest score.

Personalized PageRank for Entity Ranking In our system, we use the same dump of Wikipedia as a graph. For each mention, the teleport vector is initialized as follows: let $\{m_1, \ldots, m_N\}$ be the possible mentions found in the document, and let $A = \{a_1, \ldots, a_M\}$ be all possible candidates for the mentions $m_1, \ldots, m_N$. Note that the query string is included in the mention set. We initialize the teleport vector by assigning some value to the vertices a_i and zero to the rest.

We normalize the teleport vector so that its elements sum up to one and apply Personalized PageRank using the UKB package[20]. After Personalized PageRank computation, we output the final ranks of the articles which are the possible entity candidates of the query string.

The PageRank algorithm needs some parameters to be specified, namely the so-called damping factor and the convergence threshold (Page et al., 1999). Following usual practice, we used a damping value of 0.85 and finish the calculations after 30 iterations (Agirre and Soroa, 2009). We did not tune these parameters.

We considered two alternatives of the Personalized PageRank algorithm as follows. In the *ppr* variant, we use all articles when initializing, including the target mention. In the *word by word* variant (*w2w*), we exclude the target mention (Agirre and Soroa, 2009) when initializing, as different meanings are collapsed. For instance, several articles referred by the term *Washington* relate to the President and the capital. These articles reinforce each other, artificially boosting the probabilities of those articles and distorting the disambiguation results.

5.6 Experimental Results and Discussion

In this section, we describe and discuss experimental results for identifying Wikipedia links and for linking to a knowledge base.

[20] `http://ixa2.si.ehu.es/ukb/` (last accessed: 2014-12-07)

Name	Mention selection	Mention ranking	1% P	1% R	1% F	6% P	6% R	6% F
Baselines	Tokens	Position	.11	.03	.05	.11	.09	.10
	Noun phrases	Position	.23	.06	.09	.22	.12	.16
Text-based	Tokens	TextRank	.36	.10	.16	.36	.25	.29
	N-grams	TextRank	.28	.08	.12	.30	.23	.26
	Noun phrases	TextRank	.27	.07	.11	.26	.14	.18
	Tokens	tf-idf	.19	.05	.08	.16	.13	.14
	N-grams	tf-idf	.16	.04	.07	.16	.13	.15
	Noun phrases	tf-idf	.25	.06	.10	.23	.13	.17
	Tokens	Length	.12	.03	.05	.11	.09	.10
	N-grams	Length	.13	.03	.05	.13	.10	.11
	Noun phrases	Length	.23	.06	.09	.22	.12	.16
GPNM	Page titles	Length	.48	.13	.21	.31	.26	.28
ICLM	Link anchors	Mention strength	**.58**	**.16**	**.26**	**.38**	**.31**	**.34**

Table 5.3
Results of mention identification approaches on the Wikipedia test collection. Best results are marked bold. The text-based approaches are unsupervised approaches because they do not require any training data. ICLM is a supervised approach because it requires existing links.

5.6.1 Identifying internal Wikipedia links

We will first show our results for the subtask of mention identification and then discuss results for the subtask of entity linking. We will investigate to what extent link-based and title-based approaches are influenced by the amount of training data.

Mention Identification Results

We first analyze mention identification results for the position baselines, combinations of text-based approaches, and state of the art approaches in Table 5.3. The overall precision of all approaches is rather low. However, it is known that using Wikipedia for evaluation underestimates the actual precision of unsupervised approaches (Huang et al., 2009b), as Wikipedia contains many anchors like dates or numbers which are not in the list of candidates for mentions and are thus not captured by text-based approaches.[21] Also, we do not require perfect precision, as the user will select valid link anchors from the highlighted set of anchor candidates. The best performing text-based approach is a combination of token candidates with co-occurrence graph-based ranking. The latter generally outperforms the other ranking approaches *length* and tf-idf.

For a linking threshold level of 1% (few links per document), the title-based (ICML) and link-based approach (GPNM) perform much better than any text-based approach. However, for a linking threshold of 6% (many links per document) the text-based approach outperforms the title-based approach, and the distance to the link-based approach is much smaller. Given these large differences between the results at the two threshold

[21] Dates very often lead to an article with a list of events on this date. From a technical perspective, these lists are articles, but one could argue that they connect articles which are—besides their date—not related.

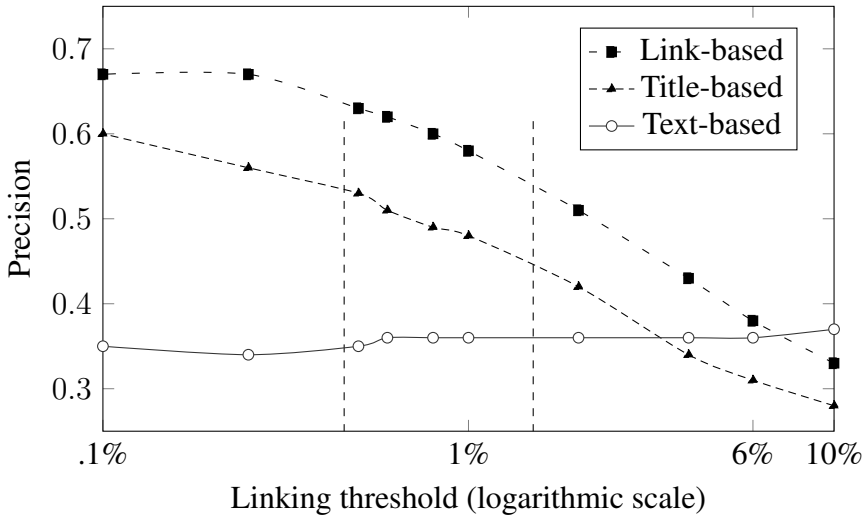


Figure 5.7: Mention identification precision depending on linking threshold (i.e. ratio of document's text that is linked). The threshold 1% and 6% are indicated by vertical lines.

levels, we systematically analyze the influence of the linking threshold on the precision of the different approaches.

Figure 5.7 shows the precision of the link-based, title-based, and the best text-based approach for different linking thresholds. It shows that link-based and title-based approaches perform well when discovering only few links (the smallest threshold is .1%, i.e. 1 in 1,000 tokens are used as anchor phrases). As the unsupervised text-based approach is independent of the linking threshold, it performs slightly better than the other approaches when discovering many links (10% or 1 in 10 tokens).

Available training data As we have seen in Table 5.3, the link-based ICML approach performs best. However, it heavily depends on the number of existing links in the collection. The large number of links in Wikipedia is clearly a special case that is due to highly motivated voluntary editors. We simulate the case of a less linked document collection by reducing the available training data in Wikipedia, thus controlling the amount of link knowledge that is used by the ICML system. We randomly remove links until only .001% of the original links ($\sim$1,000 links) are left in the training data.[22]

Figure 5.8 shows how precision changes with a decreasing number of links available for training. In contrast to link-based mention identification, text-based and title-based approaches are not influenced by the amount of available link data. Thus, they appear as horizontal lines. The link-based approach performs best, when all the training data (over 130 million links) is available, but quickly drops below the text-based approach (at around 50% of training data) and finally also below the title-based approach (at 1% of training data). As the performance of link-based approaches deteriorates rather quickly, they cannot be used to reliably predict link anchors in most other document collections, where less training data is available.

[22]Randomly removing links is only a rough approximation of the real growth (shrinkage) process of Wikipedia that can be modeled by preferential attachment (Capocci et al., 2006).

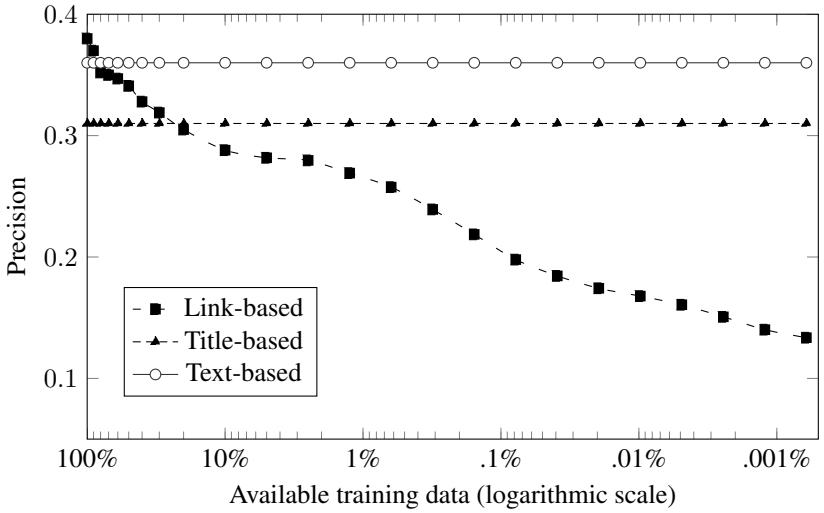


Figure 5.8: Precision of link-based mention identification, depending on the available training data at 6% threshold.

Domain transfer As we have seen above, the link-based approach only works well if a large number of links is available for training, which is hardly the case for most document collections. An obvious solution would be to use the knowledge about links and document titles from Wikipedia to improve anchor detection in other document collections. However, this turns into an issue of domain transfer, and will not work in many cases. For example, title-based mention identification uses the list of all articles in a collection as candidate anchors. Applying the list of all Wikipedia articles to another collection may not capture domain-specific anchors. For example, Wikipedia does not contain an article for each university professor. Thus, in a document collection about one specific university, the knowledge from Wikipedia will not be useful to select an anchor candidate for a link to the professors' personal homepages.

Likewise, by using link-based mention identification it is not possible to capture domain-specific anchor phrases that do not occur as a link in the training data. For example, if the link information from Wikipedia is used to discover anchors in a corporate environment, names of a company's products can probably not be discovered, as product names in Wikipedia are usually not considered worth linking, except for very well known products.

Entity linking results

Figure 5.9 shows the accuracy of the text-based and link-based approaches depending on the size of the result set. Note, that the two state of the art approaches GPNM and ICLM are both treated as link-based approaches, as they use the same steps for target identification. As we can see in Figure 5.9, the link-based approach outperforms the text-based approach for larger result sets, but they perform comparably for very small result sets. If we only consider the single top-ranked target document, the accuracy is rather low (around 50%), i.e. in only 50% of all cases can the correct target document be found on the first rank. However, if we return 10 target suggestions (which we expected to be a good size of the result set), the performance of the text-based approach improves to 70%, and that of the link-based approach to 80%. Further experiments show that accuracy rises

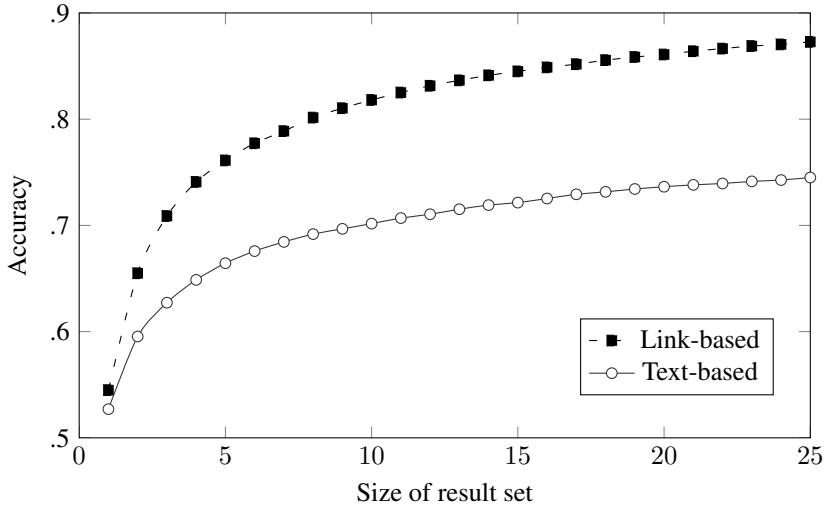


Figure 5.9: Accuracy of target identification depending on the size of the result set. An approach correctly identified a target document if it is contained in the result set.

nearly linear for more than 10 targets, but stays below 90% even for a result set size of 1,000. However, users are not expected to view such a large number of results.

Available training data We further analyzed the influence of the available amount of link knowledge by randomly reducing the number of links used for training. Figure 5.10 shows the accuracy of target identification approaches depending on the amount of available training data. The results are very similar to those for mention identification. The link-based approach outperforms the text-based approach, when all the training data (over 130 million links) is available, but drops below the text-based approach if the amount of training data is reduced. As a consequence, a large number of links is required to yield acceptable performance using the link-based approach. This means that adding a few links does not help, which makes the approach vulnerable to the *slow-start* or "cold-start" problem. As the text-based approach is not affected by a low number of links, it can provide link suggestions even for document collections without existing links.

Domain transfer Knowledge about already existing links in Wikipedia is very useful for creating new links in Wikipedia. However, it will not help much for other document collections, as can be shown using a simple example. Inside Wikipedia, it is valuable knowledge to know that the anchor phrase *Java 5* almost always points to the article about the programming language *Java*. However, this does not help us to decide in other document collections, where there might be no such document or in collections where there are more specific documents. In a document collection about programming languages, there probably exists one page for every version of Java. This cannot be captured by using the knowledge derived from Wikipedia.

5.6.2 Linking to a knowledge base (TAC-KBP)

We will first investigate results obtained using Personalized PageRank to rank entities. We tested our entity linking approach on one of the most used datasets which are freely

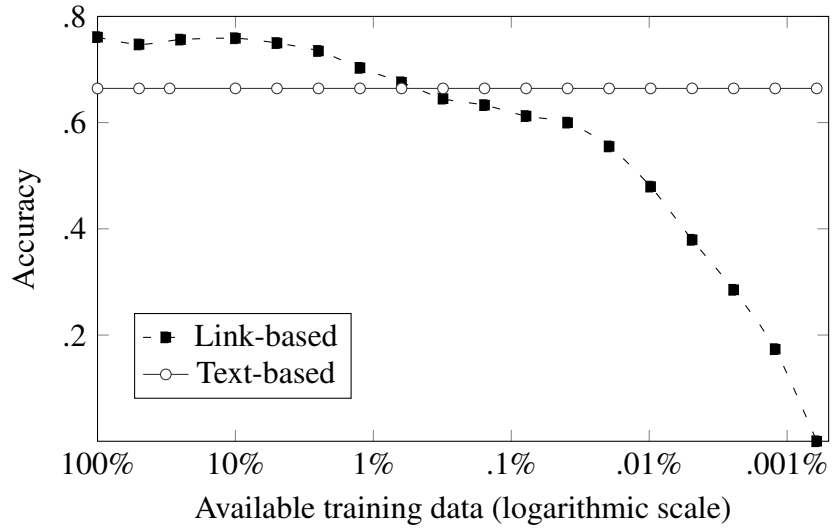


Figure 5.10: Accuracy of target identification depending on the available training data (result set size = 5).

Algorithm	Parameters	Accuracy
Upper bound		.832
MFS baseline		.610
ppr	tuned	.632
w2w	tuned	**.725**

Table 5.4
Further development results for entity linking (TAC-KBP 2009), testing different methods for using our dictionary and graphs.

available. The Knowledge Base Population shared task at the Text Analysis Conference (McNamee et al., 2010) has released evaluation datasets since 2009. The datasets comprise a set of queries, where each query consists of a document ID, a named entity string which occurs at least once in that document, and the entity ID of the correct instance in the knowledge base, which is a subset of the 2008 dump of Wikipedia. The 2009 and 2010 datasets have been used most in the entity linking literature. We use the 2009 dataset for development and 2010 for testing.

Table 5.4 compares the results of *ppr* using the best parameters on the development dataset with the rest of our implementations, including the *w2w* variant of *ppr*. We also include the *MFS* baseline, which simply returns the highest-scoring article for a mention according to the dictionary. The results show that the *w2w* variant yields a large boost, well above the other methods, and is key to obtain good results with *ppr*. All differences in respect to the other algorithms are statistically significant ($p < .001$) using McNemar's test (McNemar, 1947).

Finally, Table 5.5 shows our results on the test datasets (TAC-KBP 2010), confirming that *w2w* is key for obtaining good results. All differences of *w2w* with other algorithms are statistically significant ($p < .001$). The results on TAC-KBP 2010 are higher than for TAC-KBP 2009, due to more mentions being disambiguated to the dominant entity.

Approach	Accuracy
Upper bound	.862
MFS baseline	.612
Bunescu (2006)	.691
Milne and Witten (2008b)	.509
Varma et al. (2009)	.704
Cucerzan (2007)	.784
Hachey et al. (2011)	.798†
Lehmann et al. (2010)	**.806**
w2w	**.796**

Table 5.5
Comparison to state of the art on TAC-KBP 2010 as implemented by Hachey et al. (2011). † marks best results among several variants. Two best results are in bold, excluding those with †.

The bottom row in Table 5.5 shows our results on the test dataset (TAC-KBP 2010) with the most relevant approaches, including the best published results. Note that Hachey et al. (2011) report several systems, and we report the best results (the worst system would score 78.0 on TAC-KBP 2010). Excluding those systems, our algorithm is second best in TAC-KBP 2010, below Lehmann et al. (2010). All top scoring systems use a complex mixture of information sources which include supervised machine learning, as we will show in the following section. All in all, the good results of the method are remarkable, given the simplicity of the information source and algorithm, and the lack of fine tuning to the task. No other system relies solely on the hyperlink graph.

5.7 Computing Similarities with Senses

Previously, we have analyzed and evaluated existing approaches to link identification. In this section, we describe how links can be used for computing similarity of words, respectively senses. This is a special case of entity linking, where two entities share the same name, e.g. *Bass* can be either the instrument or the fish. Depending on their meaning, they have a high, low respectively, similarity to the entity *guitar*. Being able to compute similarity on a sense-level improves approaches to entity linking, which make use of similarities. Measuring similarity between words is also a very important task within NLP applications such as text summarization (Barzilay and Elhadad, 1997), question answering (Lin and Pantel, 2001), and automatic essay grading (Attali and Burstein, 2006). However, most of the existing approaches compute similarity on the word-level instead of the sense-level. Consequently, most evaluation datasets have so far been annotated on the word-level, which is problematic as annotators might not know some infrequent senses and are biased towards the more probable senses. In this section, we provide evidence that this fact heavily influences the annotation process. For example, when people are presented the word pair *jaguar–gamepad* only few people know that *jaguar* is also the name of an Atari game console.[23] People know the more common senses of *jaguar*, i.e.

[23]The Atari Jaguar was a popular game console in the 1990s.

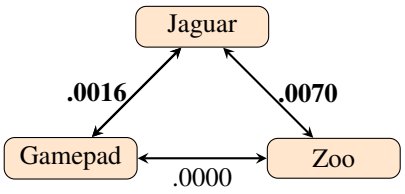


Figure 5.11: Similarity between words.

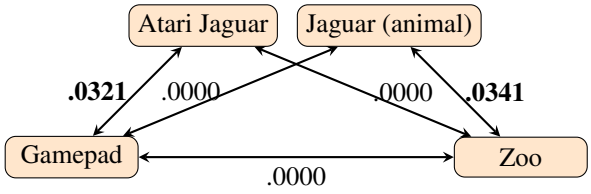


Figure 5.12: Similarity between senses.

the car brand or the animal. Thus, the word pair receives a low similarity score, while computational measures are not so easily affected by popular senses. It is thus likely that existing evaluation datasets give a wrong picture of the true performance of similarity measures on real-life data.

We analyze similarity measures, in particular, we investigate the difference between computing similarity on word-level and on sense-level. Thus, we present the means to convert a state of the art word similarity measure into a sense similarity measure. Being able to compute similarity on the sense-level is an enabling technology for text structuring. For text structuring with links, for example, we need to be able to compute how similar an anchor sense and a target sense are.

In order to evaluate the new measure, we create a special sense similarity dataset and re-rate an existing word similarity dataset using two different sense inventories from WordNet and Wikipedia. We discover that word-level measures were not able to differentiate between different senses of one word, while sense-level measures actually increase correlation when shifting to sense similarities. Sense-level similarity measures improve when evaluated with a re-rated sense-aware gold standard, while the correlation with word-level similarity measures decreases.

Thus, in this section we investigate whether similarity should be measured on the sense-level. We analyze state of the art methods and describe how the word-based Explicit Semantic Analysis (ESA) measure (Gabrilovich and Markovitch, 2007) can be transformed into a sense-level measure. We create a sense similarity dataset, where senses are clearly defined and evaluate similarity measures with this novel dataset. We also re-annotate an existing word-level dataset on the sense-level in order to study the impact of sense-level computation of similarity.

5.7.1 Word-level vs. sense-level approaches to computing similarity

Overall, existing measures either compute similarity (i) on the word-level (Finkelstein et al., 2002; Gabrilovich and Markovitch, 2007), or (ii) on the sense-level (Lin, 1998; Milne, 2007). Similarity on the word-level may cover any possible sense of the word,

whereas on the sense-level only the actual sense is considered. We use Wikipedia Link Measure (Milne, 2007) and Lin (Lin, 1998) as examples of sense-level similarity measures[24] and ESA as the prototypical word-level measure.[25]

The Lin measure is a widely used taxonomy-based similarity measure from a family of similar approaches (Budanitsky and Hirst, 2006; Seco et al., 2004; Banerjee and Pedersen, 2002; Resnik, 1999; Jiang and Conrath, 1997; Grefenstette, 1992). It computes the similarity between two senses based on the information content[26] (IC) of the lowest common subsumer (lcs) and of both senses (see Formula 5.4).

$$\text{sim}_{\text{lin}} = \frac{2\ IC(lcs)}{IC(sense1) + IC(sense2)} \tag{5.4}$$

Another type of sense-level similarity measure is based on Wikipedia that can also be considered a sense inventory, similar to WordNet. Milne (2007) uses the link structure obtained from articles to count the number of shared incoming links of articles. Milne and Witten (2008b) give a more efficient variation for computing similarity (see Formula 5.5) based on the number of links for each article, shared links $|A \cap B|$ and the total number of articles in Wikipedia $|W|$. We refer to this similarity measure as Wikipedia Link Measure (WLM).

$$\text{sim}_{\text{WLM}} = \frac{\log max(|A|, |B|) - \log|A \cap B|}{\log|W| - \log min(|A|, |B|)} \tag{5.5}$$

All sense-level similarity measures can be converted into a word similarity measure by computing the maximum similarity of all possible sense pairs. Formula 5.6 shows the heuristic, with S_n being the possible senses for word n, sim_w the word similarity, and sim_s the sense similarity.

$$\text{sim}_w(w_1, w_2) = \max_{s_1 \in S_1, s_2 \in S_2} \text{sim}_s(s1, s2) \tag{5.6}$$

Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007) is a widely used word-level similarity measure based on Wikipedia as a background document collection. ESA constructs an n-dimensional space, where n is the number of articles in Wikipedia. A word is transformed to a vector with the length n. Values of the vector are determined by the term frequency in the corresponding dimension, i.e. in a certain Wikipedia article. The similarity of two words is then computed as the inner product (usually the cosine) of the two word vectors.

The work by Schwartz and Gomez (2011) is the closest to our approach in terms of sense annotated datasets. They compare several sense-level similarity measures based on the WordNet taxonomy on sense-annotated datasets. For their experiments, annotators were asked to select senses for every word pair in three similarity datasets. Annotators were not asked to re-rate the similarity of the word pairs, or the sense pairs, respectively. Instead, similarity judgments from the original datasets are used. Possible senses are given by WordNet and the authors report an inter-annotator agreement of .93 for the Rubenstein and Goodenough (1965) dataset.

[24] We selected these measures because they are intuitive, but still among the best performing measures.

[25] Hassan and Mihalcea (2011) classify these measures as corpus-based and knowledge-based.

[26] The information content in WordNet is the logarithm of the proportion of entities below the current entity.

The authors then compare Spearman's rank correlation between human judgments and judgments from WordNet-based similarity measures. They focus on differences between similarity measures using the sense annotations and the maximum value for all possible senses. The authors do not report improvements across all measures and datasets. Of ten measures and three datasets using sense annotations, results improved in nine cases. In 16 cases, results are higher when using the maximum similarity across all possible senses. In five cases, both measures yield equal correlation. These experiments show that switching from words to senses has an effect on the performance of similarity measures.

The work by Hassan and Mihalcea (2011) is the closest to our approach in terms of similarity measures. They introduce Salient Semantic Analysis (SAS), which is a sense-level measure based on links and disambiguated senses in Wikipedia articles. They create a word-sense-matrix and compute similarity with a modified cosine metric. However, they apply additional normalization factors to optimize the evaluation metrics, which makes a direct comparison of word-level and sense-level similarity measures difficult.

Meyer and Gurevych (2012a) analyze verb similarity using a corpus from Yang and Powers (2006) and based on the work by Zesch et al. (2008b). They apply variations of the similarity measure ESA by Gabrilovich and Markovitch (2007) using Wikipedia, Wiktionary, and WordNet. Meyer and Gurevych (2012a) report improvements using a disambiguated version of Wiktionary. They automatically disambiguate links in Wiktionary articles and thus transform the resource into a sense-based one. In contrast to our work, they focus on the similarity of verbs (instead of nouns). They also apply disambiguation to improve the underlying resource. We switch the level, which is processed by the measure, to senses. We now show how ESA can be successfully adapted to also work on the sense-level.

5.7.2 DESA

In the standard definition, ESA computes the term frequency based on the number of times a term—usually a word—appears in a document. In order to apply it on the sense-level, we will need a large sense-disambiguated corpus. Such a corpus could be obtained by performing word sense disambiguation on all words (Agirre and Edmonds, 2006; Navigli, 2009). However, this is an error-prone task and requires large computational power for a big corpus. Thus, we rely on Wikipedia as an already manually disambiguated corpus. Wikipedia is a highly linked resource and articles can be considered as senses.[27] We extract all links from all articles, with the link target as the term. Figure 5.13 shows an example of a Wikipedia article with links, e.g. the term *Irish* links to a page about the Irish language and *zoo* links to an article about zoos. This approach is not restricted to Wikipedia, but can be applied to any resource containing connections between articles, such as Wiktionary (Meyer and Gurevych, 2012b). Another reason to select Wikipedia as a corpus is that it will allow us to directly compare similarity values with the Wikipedia Link Measure (WLM) as described above.

After this more high-level introduction, we now focus on the mathematical foundation of ESA and disambiguated ESA (called DESA). ESA and DESA count the frequency of each term (or sense) in each document. Table 5.6 shows the corresponding term-document matrix for the example in Figure 5.11. The term *Jaguar* appears in all shown documents,

[27] Wikipedia also contains pages with a list of possible senses called *disambiguation pages*, which we filter out.

Dublin Zoo

From Wikipedia, the free encyclopedia

Coordinates: 53°21′14″N 6°18′14″W

Dublin Zoo (Irish: ***Zú Bhaile Átha Cliath***[2][3]), in Phoenix Park, Dublin, Dublin Zoo is the largest zoo in Ireland and one of Dublin's most popular attractions. Opened in 1831,[4] the zoo describes its role as conservation, study, and education. Its stated mission is to "work in partnership with zoos worldwide to make a significant contribution to the conservation of the endangered species on Earth".

Dublin Zoo

Dublin Zoo entrance

Date opened	1 September 1831

Figure 5.13: The beginning of the Wikipedia article about Dublin Zoo. Obtained from `http://en.wikipedia.org/wiki/Dublin_Zoo` (last accessed: 2014-09-10).

Articles	**Terms**		
	Jaguar	*Gamepad*	*Zoo*
# articles	3,496	30	7,553
Dublin Zoo	1	0	25
Wildlife Park	1	0	3
D-pad	1	0	0
Gamepad	4	1	0
...	...	...	...

Table 5.6
Term-document-matrix for frequencies in a corpus.

but the term *Zoo* appears in the articles *Dublin Zoo* and *Wildlife Park*. A manual analysis shows that *Jaguar* appears with different senses in the articles *D-pad*[28] and *Dublin Zoo*.

By comparing the vectors without any modification, we see that the word pairs *Jaguar–Zoo* and *Jaguar–Gamepad* have vector entries for the same document, thus leading to a non-zero similarity. Vectors for the terms *Gamepad* and *Zoo* do not share any documents, thus leading to a similarity of zero.

Wikipedia articles are the equivalence of senses, and links between articles are the frequencies in the term-document-matrix. Shifting from words to senses changes term frequencies in the term-document-matrix of Table 5.7. The word *Jaguar* is split into the senses *Atari Jaguar* and *Jaguar (animal)*. Overall, the term-document-matrix for the sense-based similarity has lower frequencies, usually zero or one because in most cases one article does not link to another article or exactly once. Both senses of *Jaguar* do not appear in the same document, hence, their vectors are orthogonal. The vector for the term *Gamepad* differs from the vector for the same term in Table 5.6. This is due to two

[28] A D-pad is a directional pad for playing computer games.

Articles	**Senses**			
	Atari Jaguar	*Gamepad*	*Jaguar (animal)*	*Zoo*
# articles	156	86	578	925
Dublin Zoo	0	0	2	1
Wildlife Park	0	0	1	1
D-pad	1	1	0	0
Gamepad	1	0	0	0
...	...	...	...	...

Table 5.7
Sense-document-matrix for frequencies in a corpus.

effects: (i) There is no link from the article *Gamepad* to itself, but the term is mentioned in the article, and (ii) there exists a link from the article *D-pad* to *Gamepad*, though using another term.

The term/sense-document-matrices in Table 5.6 and 5.7 show unmodified frequencies of the terms. When comparing two vectors, both are normalized in a prior step. Values can be normalized by the inverse logarithm of their document frequency. Sense frequencies can also be normalized by weighing them with the inverse frequency of links pointing to an article (documents or articles with many links pointing to them receive lower weights as documents with only few incoming links.) We normalize vector values with the inverse logarithm of article frequencies.

Besides comparing two vectors by measuring the angle between them (cosine), we also experiment with a language model variant. In the language model variant, we calculate for both vectors the ratio of terms they both share.[29] The final similarity value is the average of both vectors. This is somewhat similar to the approach of Wikipedia Link Measure by Milne (2007). Both rely on Wikipedia links and are based on frequencies of these links. We show that—although, ESA and Link Measure seem to be very different—they both share a general idea and are identical with a certain configuration.

Relation to the Wikipedia Link Measure

Link Measure counts the number of incoming links to both articles and the number of shared links. In the originally presented formula by Milne (2007), the similarity is the cosine of vectors for incoming or outgoing links from both articles. Incoming links are also shown in term-document-matrices in Table 5.6 and 5.7, thus providing the same vector information. In Milne (2007), vector values are weighted by the frequency of each link normalized by the logarithmic inverse frequency of links pointing to the target. This is one of the earlier described normalization approaches. Thus, we argue that the Wikipedia Link Measure is a special case of our more general DESA approach.

[29] In the language model variant, a word is represented by a list of articles in which a term appears. Two vectors are compared by computing the overlap.

5.7.3 Annotation Study I: Rating Sense Similarity

We argue that human judgment of similarity between words is influenced by the most probable sense. We create a dataset with ambiguous terms and ask two annotators[30] to rank the similarity of senses and evaluate similarity measures with the novel dataset.

Constructing an Ambiguous Dataset

In this section, we discuss how an evaluation dataset should be constructed in order to correctly asses the similarity of two senses. Typically, evaluation datasets for word similarity are constructed by letting annotators rate the similarity between both words without specifying any senses for these words. It is common understanding that annotators judge the similarity of the combination of senses with the highest similarity.

We investigate this hypothesis by manually constructing a new dataset consisting of 105 ambiguous word pairs. Word pairs are constructed by adding one arbitrary word with two clearly distinct senses and a second word, which is very similar to only one of the senses. We first ask the annotators to rate the 105 word pairs on a scale from 0 (not similar at all) to 4 (almost identical). In the second round, we ask the same annotators to rate the same dataset, but now on a sense-level. The 277 sense[31] pairs for these word pairs are annotated using the same scale.

The final dataset thus consists of two levels: (i) word similarity ratings, and (ii) sense similarity ratings. The gold ratings are the averaged ratings of both annotators, resulting in a Krippendorf's α agreement[32] of .510 (Spearman: .476) for word ratings and .792 (Spearman: .784) for sense ratings. A Krippendorf's α above .8 is considered reliable and below .667 discarded. Additionally, we report the correlation of the human annotators in terms of Spearman. Annotations on the sense-level can thus be considered, while annotations on the word-level seem to be too hard. Spearman rank correlation is a metric based on the ranks of distributions. Disagreements are due to unknown senses of words and differences in the magnitude of similarity.

Table 5.8 shows ratings of both annotators for two word pairs and ratings for all sense combinations. In the given example, the word *bass* has the senses of the fish, the instrument, and the sound. Annotators compare the words and senses of the words *Fish* and *Horn*, which appear only in one sense (most frequent sense) in the dataset.

The annotators' rankings contradict the assumption that the word similarity equals the similarity of the sense pairs with highest sense similarity. Instead, the highest sense similarity rating is higher than the word similarity rating. This may be caused—among others—by two effects: (i) the correct sense is not known or not recalled, or (ii) the annotators (unconsciously) adjust their ratings to the probability of the sense. Although, the annotation manual stated that Wikipedia (the source of the senses) could be used to get informed about senses and that any sense for the words can be selected, we see both effects in the annotators' ratings. Both annotators rated the similarity between *Bass* and *Fish* as very low (1 and 2). However, when asked to rate the similarity between the sense *Bass (Fish)* and *Fish*, both annotators rated the similarity as high (4). Accordingly, word

[30] Annotators are near-native speakers of English and have a university degree in cultural anthropology and computer science.

[31] The sense of a word is given in parentheses but annotators have access to Wikipedia to get information about those senses.

[32] We report agreement as Krippendorf's α with a quadratic weight function. (Krippendorff, 2012)

$Word_1$	$Word_2$	$Sense_1$	$Sense_1$	$Annotator_1$ Words	$Annotator_1$ Senses	$Annotator_2$ Words	$Annotator_2$ Senses
Bass	**Fish**	Bass (Fish)	Fish (Animal)	**1**	**4**	**1**	**4**
		Bass (Instrument)			1		1
		Bass (Sound)			1		1
Bass	**Horn**	Bass (Fish)	Horn (Instrument)	**2**	1	**1**	1
		Bass (Instrument)			**3**		**4**
		Bass (Sound)			**3**		3

Table 5.8
Examples of ratings for two word pairs and all sense combinations with the highest ratings marked bold.

	Measure	Word-level Spearman	Word-level Pearson	Sense-level Spearman	Sense-level Pearson
Human upper bound[34]		*.476*	*.471*	*.784*	*.793*
Word measures	ESA	**.456**	.239	-.001	.017
	Lin (WordNet)	.298	.275	.038	.016
Sense measures	DESA (Cosine)	.292	.272	**.642**	.348
	DESA (Lang. Mod.)	.185	.256	**.642**	**.482**
	WLM (out)	.190	.193	.537	.372
	WLM (in)	.287	**.279**	.535	.395

Table 5.9
Correlation of similarity measures with a human gold standard on ambiguous word pairs. The improvement of all sense-level similarity measures over ESA is significant at $p \leq .001$.

similarity for the word pair *Bass* and *Horn* is low (1), but their sense similarity is higher (3 and 4).

Results & Discussion

We evaluated similarity measures with the previously created new datasets. Table 5.9 shows correlations of similarity measures with human ratings. We divide the table into measures computing similarity on word-level and on sense-level. ESA works entirely on a word-level, Lin (WordNet) uses WordNet as a sense inventory, which means that senses differ across sense inventories.[33] DESA and Wikipedia Link Measure (WLM) compute similarity on the sense-level, however, similarity on the word-level is computed by taking the maximum similarity of all possible sense pairs.

Results in Table 5.9 show that word-level measures return the same rating independent from the sense being used, thus, they perform well when evaluated on the word-level, but perform poorly on the sense-level. For the word pair *Jaguar–Zoo*, there exist two sense pairs *Atari Jaguar–Zoo* and *Jaguar (animal)–Zoo*. Word-level measures return the

[33] Although, there exist sense alignment resources, we did not use any alignment.
[34] The human upper bound is defined as the respective correlation between both human annotators.

same similarity, thus leading to a very low correlation. This was expected, as only sense-based similarity measures can discriminate between different senses of the same word. Somewhat surprisingly, results for sense-level measures are also good on the word-level, but increase strongly on the sense-level. Our novel measure DESA provides the best results. This is expected as the ambiguous dataset contains many infrequently used senses, which annotators are not always aware of. An interview of the annotators about their decisions showed that many senses are known to the annotator (e.g. the fish sense of *Bass*), however, the annotator did not consider them when annotating the similarity of the word pair *Bass–Fish*.

Our analysis shows that the algorithm for comparing two vectors (i.e. cosine and language model) only has an influence on the results for DESA when computed on the word-level. The main difference between both ways to compare vectors is that cosine comparison considers the frequency of a term, while the language model operates on the bag-of-words. In case of very sparse term-document-matrices, many ties lead to a lower Spearman rank correlation. Correlation for the Wikipedia Link Measure (WLM) differs depending on whether the overlap of incoming or outgoing links[35] is computed. In comparison to that, articles with few outgoing links (e.g. short articles) have a high similarity (if they share any of the few links) or a similarity of zero (if they do not share any links). Milne and Witten (2008b) chose to use incoming links for performance reasons. WLM on word-level using incoming links performs better, while the difference on sense-level evaluation is only marginal. This might be due to higher fluctuations of the correlation. On word-level, 105 word pairs are compared, on the sense-level 277 sense pairs yield a better empirical basis.

In order to evaluate the significance of correlations, we use the equation proposed by Press (2007, p. 745). Press says, that Pearson's correlation "[...] is a rather poor statistic for deciding *whether* one observed correlation is statistical significant and/or whether one observed correlation is significantly stronger than another. The reason is that r is ignorant of the individual distributions of x and y, so there is no universal way to compute its distribution in the case of the null hypothesis." The same applies for Spearman rank correlation as it does not include the actual distribution of x and y. However, Press (2007) presents an equation (see Equation 14.5.10) to compute statistical significance of correlation using the Fisher's z-transformation (Fisher, 1915, 1921) and the complementary error function (Greene, 2003).[36] All improvements for the sense-level dataset obtained by sense measures are significant at $p \leq .001$.

Overall results show that only sense-level similarity measures can differentiate between different senses of word pairs and thus compute the similarity of sense pairs. On the word-level, the ESA word-level measure outperforms all other measures in respect to Spearman. However, it is relevant to investigate how well a measure can decide which senses of word pairs have a higher similarity. Especially for entity linking, it needs to be decided which sense is best for the other (less ambiguous) words from the context.

Min. diff.	#pairs	measure	Correct	Wrong: Reverse	Wrong: Values equal	Accuracy
0.5	*382*	*Human upper bound*[37]	*334*	*8*	*40*	*.87*
	420	DESA	296	44	80	.70
		WLM (in)	296	62	62	.70
		WLM (out)	310	76	34	**.74**
1.0	*382*	*Human upper bound*	*334*	*8*	*40*	*.87*
	390	DESA	286	38	66	.73
		WLM (in)	282	52	56	.72
		WLM (out)	294	64	32	**.75**
1.5	*338*	*Human upper bound*	*309*	*3*	*26*	*.91*
	360	DESA	264	34	62	.73
		WLM (in)	260	48	52	.72
		WLM (out)	280	54	26	**.78**
2.0	*338*	*Human upper bound*	*309*	*3*	*26*	*.91*
	308	DESA	232	28	48	.75
		WLM (in)	226	36	46	.73
		WLM (out)	244	46	18	**.79**
2.5	*241*	*Human upper bound*	*224*	*2*	*15*	*.93*
	280	DESA	216	22	42	.77
		WLM (in)	206	32	42	.74
		WLM (out)	224	38	18	**.80**
3.0	*241*	*Human upper bound*	*224*	*2*	*15*	*.93*
	174	DESA	134	10	30	.77
		WLM (in)	128	20	26	.74
		WLM (out)	136	22	16	**.78**
3.5	*49*	*Human upper bound*	*48*	*1*	*0*	*.98*
	68	DESA	56	4	8	**.82**
		WLM (in)	50	6	12	.74
		WLM (out)	52	6	10	.76
4.0	*49*	*Human upper bound*	*48*	*1*	*0*	*.98*
	12	DESA	10	2	0	**.83**
		WLM (in)	10	2	0	**.83**
		WLM (out)	10	2	0	**.83**

Table 5.10
Pair-wise comparison of sense pairs for several measures. We count the number of cases where the sense pair with the higher similarity is correctly identified. This count divided by the total number of cases is defined as accuracy. We evaluate with a different selection of the ambiguous dataset considering only those cases, where human annotators rated the similarity of two sense pairs with a minimal difference. A naïve baseline (selecting the first pair) reaches an accuracy of .5.

Pair-wise Evaluation

[35] Milne (2007) originally proposed to use weighted outgoing links, but they apply this approach for both directions in Milne and Witten (2008a).

[36] An implementation can be found in DKPro Statistics (Meyer et al., 2014).

In a second experiment, we evaluate how well sense-based measures can decide, which one of two sense pairs for one word pair has a higher similarity. We thus create for every word pair all possible sense pairs[38] The word pair *Bank–Bond* will be expanded to the sense pairs *Bank (finance)–Bond (finance)* and *Bank (geography)–Bond (finance)*, of which only the first one is similar. Formally, for every word pair w_1–w_2, there exist at least two sense pairs $s_{1,1}$–$s_{2,1}$ ($s_{1,1}$ is the first sense for w_1 and $s_{2,1}$ is the first sense for w_2) and $s_{1,2}$–$s_{2,1}$ for which human annotators have rated the similarity of one sense pair higher than the other. For evaluation, we count the cases a measure correctly identifies the sense pair with a higher similarity. We further divide pairs into cases in which a measure identifies the wrong sense pair or identifies both sense pairs as on the same level.

Table 5.10 shows the evaluation results for the accuracy of selecting the sense pair with the higher similarity. We construct the evaluation dataset by combining all sense pairs for a word pair with the sense pairs for the same word pair. We evaluate with a selection of all possible combinations of sense pairs based on a minimal difference of similarity between two sense pairs. For example, human annotators rated the sense pair *Bank (finance)–Bond (finance)* with a similarity of 2.5 and the sense pair *Bank (geography)–Bond (finance)* with a similarity of 0. The difference of these sense pairs is thus 2.5 and it will be contained in all datasets with a minimal difference of 2.5 and below. Column *#pairs* gives the number of remaining sense pairs, which decreases for a higher minimal difference. If a measure classifies two sense pairs wrongly, it may either be because it rated the sense pairs with an equal similarity or because it reversed the order.

The Results show that the accuracy increases with a higher minimum difference between sense pairs. Figure 5.14 emphasizes this finding. Overall, accuracy for this task is high (between .70 and .83), which shows that all measures can discriminate sense pairs better than the baseline (significant at $p \leq .05$ for all differences smaller than 4.0). None of the measures reaches the upper bound for this task, which raises from .87 to .98 for a larger difference. However, all measures still outperform a naïve baseline (selecting the first pair as the one with a higher similarity) of .5. The measure WLM (out) performs best for most cases with a non-significant difference in accuracy of up to .06.

Although, DESA has the highest correlation with human ratings, it is outperformed by WLM (out) on the task of discriminating two sense pairs. An error analysis reveals that this difference is due to many *zero-similarity* cases returned by WLM (out). WLM (out) frequently returns a similarity of zero in cases when other measures still return a non-zero similarity. This harms Spearman's rank correlation (because of many ties) and Pearson's correlation (damping of the curve in the area of lower similarities), while it is beneficial for deciding which sense pair has a higher score.

When comparing these results to the results from Table 5.9, we see that a high correlation does not imply accurate discrimination of sense pairs; results are not stable across both evaluation scenarios. A measure performing well with respect to one metric does not necessarily perform equally well with respect to another metric. This is due to different characteristics of the evaluation metrics. Spearman and Pearson evaluate by computing

[37]The human upper bound is calculated by using the human rating from one annotator as the gold standard and comparing them with ratings from the other annotator. This is done for each annotator and results are averaged. The number of instances in the evaluation dataset might differ from the evaluation dataset for the measures because the gold standard is altered.

[38]For one word pair with two senses for one word, there are two possible sense pairs. Three senses result in three sense pairs.

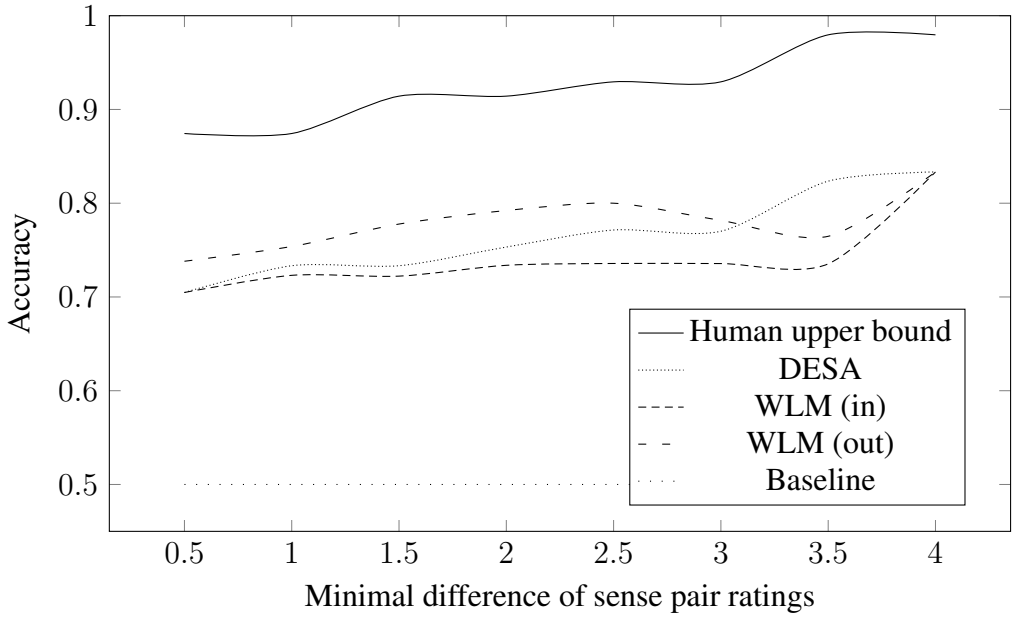


Figure 5.14: Accuracy distribution depending on difference of similarity ratings.

the correlation of the similarity ranks of the measures and the human gold standard. Spearman considers the overall ranks, while Pearson considers the absolute values. The major difference is that all pairs are evaluated at once, while in the evaluation setting as shown in Table 5.9, only two sense pairs are evaluated at a time. The absolute difference of the similarity is not considered. Hence, the selection of the evaluation metric depends on the task. In case absolute values or the ranking of many sense pairs is required, Spearman and Pearson are better suited. In case the differentiation between two sense pairs is required, accuracy of two pairs is better suited. We now evaluate similarity not only on pairs of different senses, but on a standard benchmark dataset.

5.7.4 Annotation Study II: Re-rating of RG65

We performed a second evaluation study where we asked three human annotators[39] to rate the similarity of word-level pairs in the dataset by Rubenstein and Goodenough (1965). The dataset consists of 65 common noun pairs, e.g. *automobile–car* or *autograph–shore*. In the original dataset, 51 human annotators judged the similarity of word pairs on a scale of 0 to 4. The human annotators were asked to rate the similarity on the word-level.

We hypothesize that measures working on the sense-level should have a disadvantage on word-level annotated datasets due to the effects described above, which influence annotators towards more frequent senses, thus using some kind of sense weights. In our annotation studies, our aim is to minimize the effect of sense weights.

In previous annotation studies (Rubenstein and Goodenough, 1965; Miller and Charles, 1991; Finkelstein et al., 2002), human annotators could take sense weights into account when judging the similarity of word pairs. Some senses might not be known by annota-

[39] As before, all three annotators are near-native speakers of English and have a university degree in physics, engineering, and computer science. One of the annotators from the first study also participated in this study.

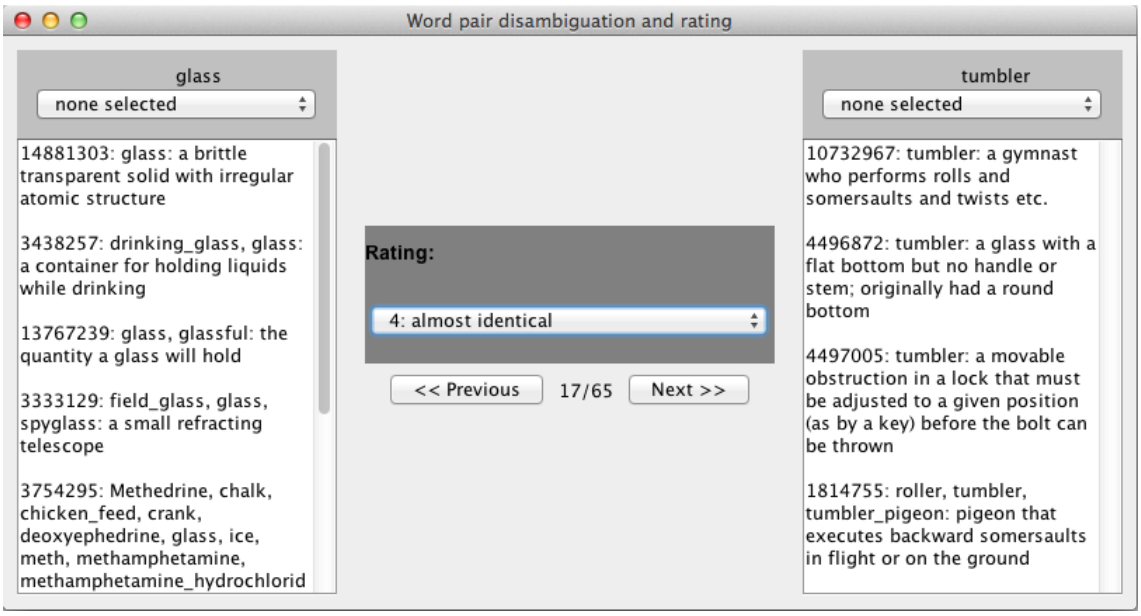


Figure 5.15: User interface for annotation studies. The example shows the word pair *glass–tumbler* with no senses selected. The interface shows WordNet definitions of possible senses in the text field below the sense selection. The highest similarity is obtained when selecting sense #4496872 for tumbler, since it is a drinking glass (sense #3438257).

tors and, thus receive a lower rating. We minimize these effects by asking annotators to first select the best sense for a word based on a short summary of the corresponding sense and then rate the similarity. To mimic this process, we created an annotation tool (see Figure 5.15), for which an annotator first selects senses for both words, which have the highest similarity. Then the annotator ranks the similarity of these sense pairs based on the complete sense definition.

A single word without any context cannot be disambiguated properly. However, when word pairs are given, annotators first select senses based on the second word. E.g. if the word pair is *Jaguar* and *Zoo*, an annotator will select the wild animal for *Jaguar*. After disambiguating, an annotator assigns a similarity score based on both selected senses. To facilitate this process, a definition of each possible sense is shown.

As in the previous experiment, similarity is annotated on a five-point-scale from 0 to 4. We ask three annotators to select senses for word pairs, but we retrieve only one similarity rating for each word pair, which is the sense combination with the highest similarity. This assures that we have a similarity rating for every pair, although annotators do not need to select the same sense for every word. In the following, we describe the annotation when using different sense inventories in the sense selection process.

No sense inventory We compare our re-ratings with the original dataset from Rubenstein and Goodenough (1965), we asked the annotators to rate similarity of word pairs without any given sense repository, i.e. comparing words directly. The annotators reached a Krippendorf's α of .73, which can be considered as a gold standard. The average Pearson's correlation of the three annotators is .82, which is close to the reported correlation of .85[40] in the original dataset (Rubenstein and Goodenough, 1965). The resulting gold stan-

[40] Average correlation of a group of 15 annotators.

dard has a high correlation with the original dataset (.923 Spearman and .938 Pearson). This is in line with our expectations, and previous work showed that similarity ratings are stable across time (Bär et al., 2011b).

Wikipedia sense inventory We now use the full functionality of our annotation tool and ask the annotators to first select senses for each word and second, rate the similarity. Possible senses and definitions for these senses are extracted from Wikipedia.[41] The same three annotators reached a Krippendorf's α of .66 (this can be considered as a substantial agreement). Comparing our gold standard obtained from the re-ratings with the original gold standard by Rubenstein and Goodenough (1965), we get a high correlation of .881 Spearman and .896 Pearson between both. This shows that our re-rating is indeed valid.

WordNet sense inventory Similar to the previous experiment, we list possible senses for each word. In this experiment, we use WordNet senses, thus, not using any named entities. The annotators reached a Krippendorf's α of .73 (this can be considered as a substantial agreement) and the resulting gold standard has a high correlation (.917 Spearman and .928 Pearson) with the original gold standard in RG65.

Comparison of ratings Figure 5.16 shows annotator ratings in comparison to similarity ratings in the original dataset. We plot the original ratings for word similarity values against the re-ratings without any sense inventory (solid line), with the Wikipedia sense inventory (dotted line), and the WordNet sense inventory (dashed line). Every data point is the average of five word similarity ratings. There is an almost linear relation between the original ratings and the re-ratings, showing the validity of the re-ratings. We observe that similarity ratings while using a sense inventory are slightly higher than ratings without any sense inventory. Annotators first select senses which have the highest similarity and then rate it. This leads to higher ratings, especially for cases where annotators selected senses other than the dominant one.

Results & Discussion

We evaluate all mentioned similarity measures using Spearman's rank correlation and Pearson's correlation[42] We calculate correlations to four human judgments: (i) from the original dataset (Orig.) (Rubenstein and Goodenough, 1965), (ii) from our re-rating study (Rerat.), (iii) from our study with senses from Wikipedia (WP), and (iv) with senses from WordNet (WN). Table 5.11 shows the results for all described similarity measures.

ESA[43] achieves a Spearman's rank correlation of .751 and a slightly higher correlation (.765) on our re-rating gold standard. Correlation then drops when compared to gold standards with senses from Wikipedia and WordNet. This is expected as the gold standard becomes more sense-aware.

Lin is based on senses in WordNet and outperforms all other measures on the original gold standard. Correlation reaches a high value for the gold standard based on WordNet,

[41] We use the English Wikipedia version from June 15^{th}, 2010.

[42] As discussed earlier Spearman's rank correlation compares the ranking of a systems output with the gold standard ranking and Pearson's correlation compares their absolute values.

[43] ESA is used with normalized text frequencies, a constant document frequency, and a cosine comparison of vectors.

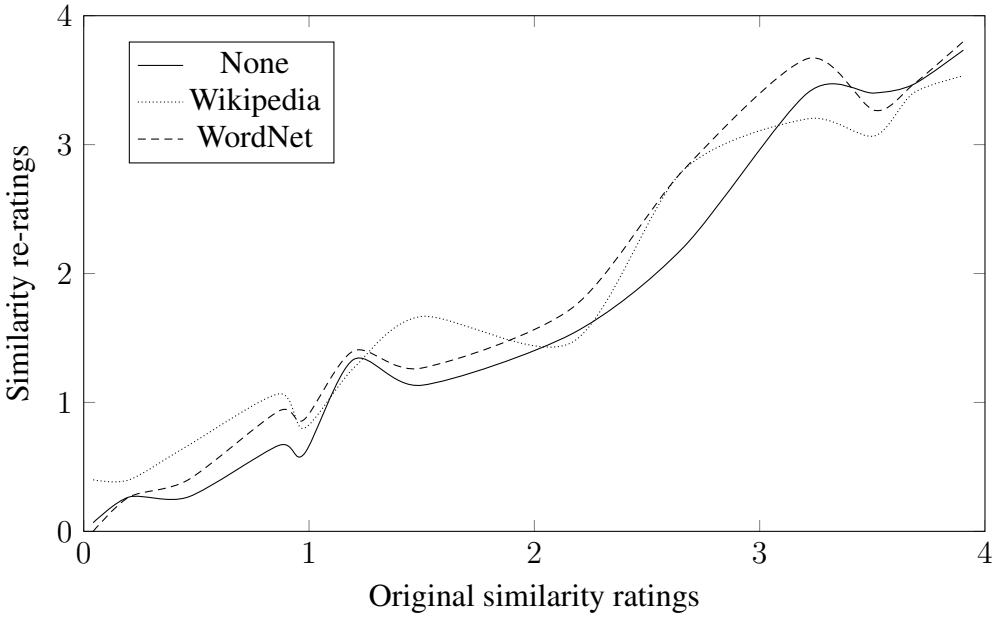


Figure 5.16: Correlation curve of re-rating studies (using the average of five adjacent data points).

	Spearman				Pearson			
Measure	Orig.	Rerat.	WP	WN	Orig.	Rerat.	WP	WN
Human upper bound	-	*.802*	*.770*	*.829*	-	*.819*	*.761*	*.833*
ESA	.751	.765	.704	.705	.647	.694	.678	.625
Lin	**.815**	.768	.705	.775	**.873**†	**.840**†	**.798**	**.846**†
DESA (lang. mod.)	.733	.765	.782	.751	.703	.739	.739	.695
DESA (cosine)	.775	**.810**	**.826**†	**.795**	.694	.712	.736	.699
WLM (in)	.716	.745	.754	.733	.708	.712	.740	.707
WLM (out)	.583	.607	.652	.599	.548	.583	.613	.568

Table 5.11
Correlation of similarity measures with a human gold standard on the word pairs by Rubenstein and Goodenough (1965). Best results for each gold standard are marked bold. Significant improvements at $p \leq .1$ compared to the very strong baseline ESA are marked with a †.

as the same sense inventory for both human annotations and measures is applied. Values for Pearson's correlation emphasize this effect: Lin reaches the maximum of .846 on the WordNet-based gold standard.

Correspondingly, the similarity measures DESA and WLM reach their maximum on the Wikipedia-based gold standard. As for the ambiguous dataset in Section 5.7.3, DESA outperforms both WLM variants. Cosine vector comparison again outperforms results of the language model variant for Spearman's rank correlation, but provides weaker results in terms of Pearson's correlation. Cosine vector comparison is a normalized metric scoring 1 for identical terms, but is negatively impacted by a sparse space as it tends to provide low scores for near synonyms (Hassan and Mihalcea, 2011). This reduces results in terms of Pearson's correltion but does not harm Spearman's rank correlation as the absolute value is not considered. As before, WLM (in) outperforms WLM (out) across

all datasets and both correlation metrics. An error analysis shows that WLM (in) returns many *zero-similarity* cases which impedes better results. DESA provides stable results, but using cosine normalization leads to a sparse distribution of similarity values. Values for similarity of near-synonyms using DESA are lower than for any variant of WLM. A manual error analysis shows that similarity results obtained with DESA are in general low, even if both terms are near synonyms. This does not affect evaluation with Spearman's rank correlation, but harms evaluation results for Pearson's correlation. Lower values are due to using the vector product for normalization. To improve results in terms of Pearson's correlation we propose to use an additional normalization factor[44] to increase lower similarity values.

Is word similarity sense-dependent? In general, sense-level similarity measures improve when evaluated with a sense-aware gold standard, while correlation with word-level similarity measures decreases. A further manual analysis shows that sense-level measures perform well when rating very similar word pairs because other senses are not considered (e.g. for the word pair *Jaguar–Gamepad* only the sense of the game console for *Jaguar* is used). This is very useful for applications such as information retrieval, where a user is only interested in very similar documents.

Our evaluation thus shows that word similarity should not be considered without accounting for the effect of the used sense inventory. The same annotators rate word pairs differently if they can specify senses explicitly (as seen in Table 5.8). Correspondingly, results for similarity measures depend on which senses can be selected. Wikipedia contains many entities, e.g. music bands or actors, while WordNet contains fine-grained senses for things (e.g. fine-grained senses of glass as shown in Figure 5.15). Using the same sense inventory as the one, which has been used in the annotation process, leads to a higher correlation.

In this section, we investigated word-level and sense-level similarity measures and investigated their strengths and shortcomings. We evaluated how correlations of similarity measures with a gold standard depend on the sense inventory used by the annotators.

We compared the similarity measures ESA (corpus-based), Lin (WordNet), and Wikipedia Link Measure (Wikipedia), and a sense-enabled version of ESA and evaluated those with a dataset containing ambiguous terms. Word-level measures were not able to differentiate between different senses of one word, while sense-level measures could even increase the correlation when shifting to sense similarities. Sense-level measures obtained accuracies between .70 and .83 when deciding which of the two sense pairs has a higher similarity.

We performed re-rating studies with three annotators based on the dataset by Rubenstein and Goodenough (1965). Annotators were asked to first annotate senses from Wikipedia and WordNet for word pairs and then judge their similarity based on the selected senses. We evaluated similarity approaches with these new human gold standards and found that the correlation depends on the resource used by the similarity measure and the sense inventory available to a human annotator. The performance of sense-level similarity measures improves when evaluated with a sense-aware gold standard, while correlation with the word-level similarity measures decreases. Using the same sense inventory as the one, which has been used in the annotation process, leads to a higher correlation. This

[44] Hassan and Mihalcea (2011) apply a normalization and report improvement.

has implications for creating word similarity datasets and evaluating similarity measures using different sense inventories: Similarity datasets should be created on a sense-level or using the same sense inventory as used for the similarity approaches.

5.8 Links in Text Structuring Scenarios

In our evaluation of link identification, we used two datasets. The Wikipedia dataset, e.g. the edition from October 8th, 2008 containing 2,666,190 articles and 135,478,255 links, is an example of a densely linked dataset containing many topics. The high number of links allows for using the internal link structure of the dataset for identifying further links. The second dataset contains web data and news texts, but also uses Wikipedia (from April 5th, 2008) as the sense inventory. The documents do not contain any links and only one designated mention should be linked to an entity.

In our evaluation for sense similarity, we used one dataset with ambiguous words and re-annotated an existing commonly used dataset. Computing the similarity of senses is a special case of entity linking, where multiple entities with the same name exist and similarity to the context is computed. This leads to improvements for all scenarios in which a large knowledge base, including ambiguous terms, exists.

Focused searcher

The focused searcher starts with an overview page of the relevant topic. She/he then uses links to browse to more specific sites. Automatically creating such links helps a focused searcher because she/he can quickly collect a list of relevant documents for the topic.

However, one needs to differentiate between links to internal sites (sites from the same domain) and links to large knowledge bases such as Wikipedia. To identify links to Wikipedia, approaches which make use of the Wikipedia link structure and Wikipedia titles yield best results. To identify links in more specialized domains, e.g. on a university lecture-level, existing links in Wikipedia cannot be used as they use a too general sense inventory. Using a much smaller number of existing internal links yields worse than using text-based approaches, as link-based approaches require a huge number of existing links. Link-based and text-based approaches perform on par with about 1% of all Wikipedia links, which are still more than 100,000 links.

Recreational news reader

A recreational news reader jumps from one interesting article to another. These *jumps* are solely motivated by the reading interests. Having many options for reading further news is helpful (and also common practice) for news agencies to make readers stick to the news page. Automatically creating such links reduces the manual effort of asking writers to create these links.

For the domain of world news, a higher recall of link identification is desired, as linked documents do not necessarily need to be highly related as readers' interest can be very broad.

Knowledge worker in a company

In corporate environments, document collections tend to be less structured (Buffa, 2006). To find information without links, one needs to perform a search to get a list of relevant documents. One can use overview documents with links pointing to more specific documents, or follow links to information, which is shared by multiple documents and thus kept in a separate document. Information can thus be divided into multiple documents and linked if necessary. The links connecting the documents can then be again used for the ranking process in information retrieval (Page et al., 1999). Thus, automatically identifying links which link from mentions in a document to another document help a knowledge worker to find information quicker.

In corporate environments, especially the *cold-start* problem is an issue. Text-based approaches can help with creating the first links and give support to workers in a company by suggesting links, which can be automatically added.

5.9 Chapter Summary

The main contributions of this chapter can be summarized as follows:

Contribution 5.1: *We presented an overview of link identification approaches and analyzed the effect of training data on the evaluation results.*
Contribution 5.2: *We used the Personalized PageRank algorithm for entity linking.*
Contribution 5.3: *We introduced and enhanced a transformation from word similarity metrics to sense similarity metrics and compared results to existing similarity metrics.*

In this chapter, we evaluated the performance of link identification approaches and presented a classification scheme for those with respect to the type of knowledge being used. We evaluated these on a test collection derived from Wikipedia, and showed that the link-based approach outperforms all other approaches if it can draw knowledge from a huge number of already existing links. However, other document collections normally contain much fewer links, and thus provide less knowledge about good link anchors and targets. As a consequence, link-based approaches suffer from the *cold-start problem*, i.e. in a collection with only a few links they do not provide helpful link suggestions.[45] Their performance only gets acceptable when a large number of links is manually added to the collection. In contrast, the text-based and title-based approaches are able to provide linking support, even if no links have been added so far.

Furthermore, we argued that knowledge from Wikipedia which is needed for title-based and link-based approaches is not necessarily transferable to other domains. Thus, text-based approaches are the best choice for reliable link identification in arbitrary document collections.

We have shown that the Personalized PageRank algorithm yields competitive results to the state of the art. This is remarkable given the fact that it only uses hyperlinks between articles, compared to more complex sources of information, including supervised methods for link identification. We also show that fine-grained optimization of parameters is not an

[45]The *cold-start problem* is the opposite of the network effect (Shapiro and Varian, 2013), which infers that a resource with few links is less likely to receive further links than a fully linked resource.

issue, as we obtain very similar values with default parameters or parameters optimized in either task.

To show the usefulness of links, we compared the similarity measures ESA (corpus-based), Lin (WordNet), and Wikipedia Link Measure (Wikipedia), and a sense-enabled version of ESA (DESA) and evaluated those with a dataset containing ambiguous word pairs and the Rubenstein Goodenough (RG65) dataset. With the ambiguous dataset, word-level measures were not able to differentiate between different senses of one word, while sense-level measures could even increase the correlation when shifting to sense similarities. We evaluated similarity approaches with the re-annotated RG65 dataset and found that the correlation heavily depends on the sense inventory used by the similarity measure and the sense inventory a human annotator had available while annotating similarity. The performance of sense-level similarity measures improves when evaluated with a sense-aware gold standard, while correlation with the word-level similarity measures decreases.

Chapter 6

Prototypes for Text Structuring Systems

In the previous chapters we have presented techniques for text structuring based on identifying keyphrases, table-of-contents, and links. A central goal when developing new technology is to make it available for users in order to support them in their everyday tasks. In this chapter, we focus on prototypes for text structuring systems as shown in Figure 6.1. We have developed two prototypes in the course of this thesis: *Wikulu* and *open window*. Wikulu (Bär et al., 2011a) is a wiki extension and mainly targeted for the corporate domain and open window is an extension to massive open online courses[1] in the educational domain. The aim of both prototypes—although used in different environments—is to connect text structuring techniques to user applications.

6.1 Wikulu

Wikis are used as collaborative information management systems (Leuf and Cunningham, 2001) and have been widely adopted in corporate and public settings (Buffa, 2006), but due to their distributed and collaborative way of construction, they suffer from a number of shortcomings. As they do not enforce their users to structure pages or add complementary metadata, wikis often end up as a mass of unmanageable pages with meaningless page titles and no usable link structure (Buffa, 2006). Over time, this leads to significant usability limitations which makes it increasingly difficult to add further content (Désilets et al., 2005).

To solve this issue, we developed the Wikulu system which uses automatic text structuring to support wiki users with their typical tasks of adding, organizing, and finding content. Support integrated in Wikulu includes *text segmentation* to segment long pages, *keyphrase extraction*, and *text summarization* to help reading long pages. Wikulu allows to integrate any NLP component which conforms to the standards of *Apache UIMA* (Ferrucci and Lally, 2004).

Wikulu is designed to integrate seamlessly with any wiki. Our system is implemented as an HTTP proxy server which intercepts the communication between the web browser and the underlying wiki engine. No further modifications to the original wiki installation

[1] Massive open online courses are lecture series organized as online material for a broader audience. `www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html` (last accessed 2014-09-01)

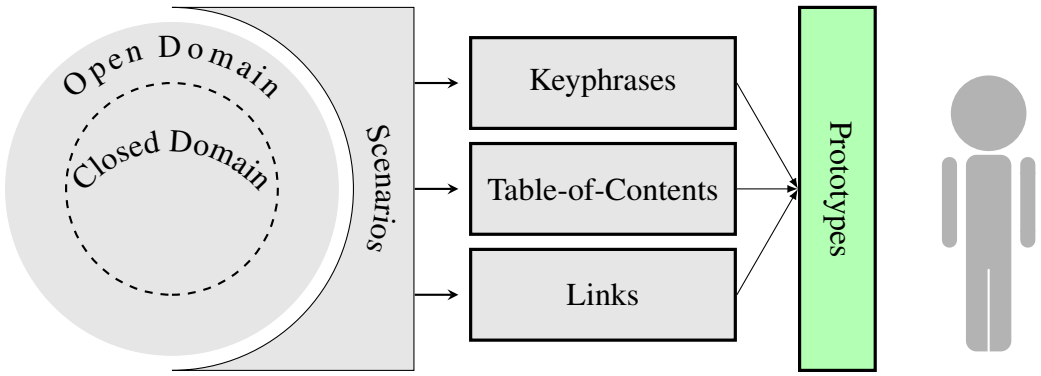


Figure 6.1: Graphical overview of the contents which are covered in this thesis with the user interface highlighted green.

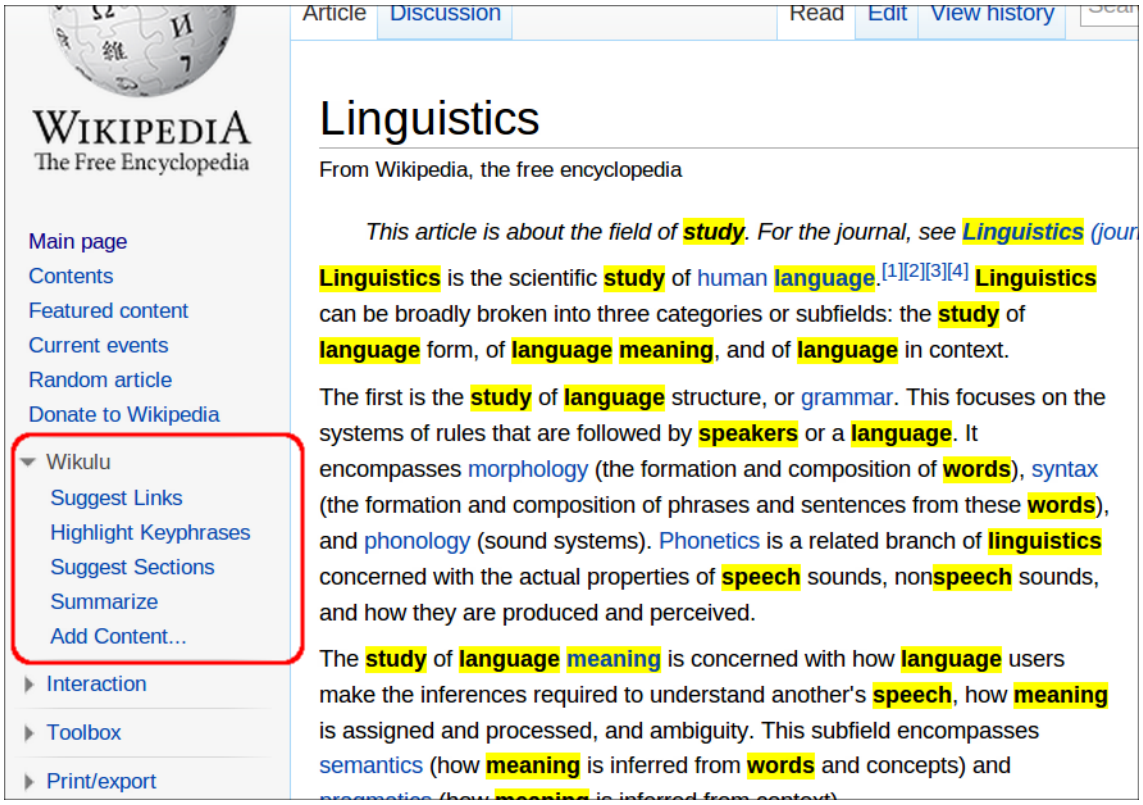


Figure 6.2: Integration of Wikulu with Wikipedia. The augmented toolbar (red box) and the results of a keyphrase extraction algorithm (yellow text spans) are highlighted.

are necessary. Currently, our system prototype contains adapters for two widely used wiki engines: *MediaWiki*[2] and *TWiki*[3]. Adapters for other wiki engines can be added with minimal effort.

In Figure 6.2 and Figure 6.3, we show the integration of Wikulu with Wikipedia and

[2] `http://mediawiki.org` (last accessed: 2014-12-07), e.g. used by *Wikipedia*
[3] `http://twiki.org` (last accessed: 2014-12-07), often used for corporate wikis

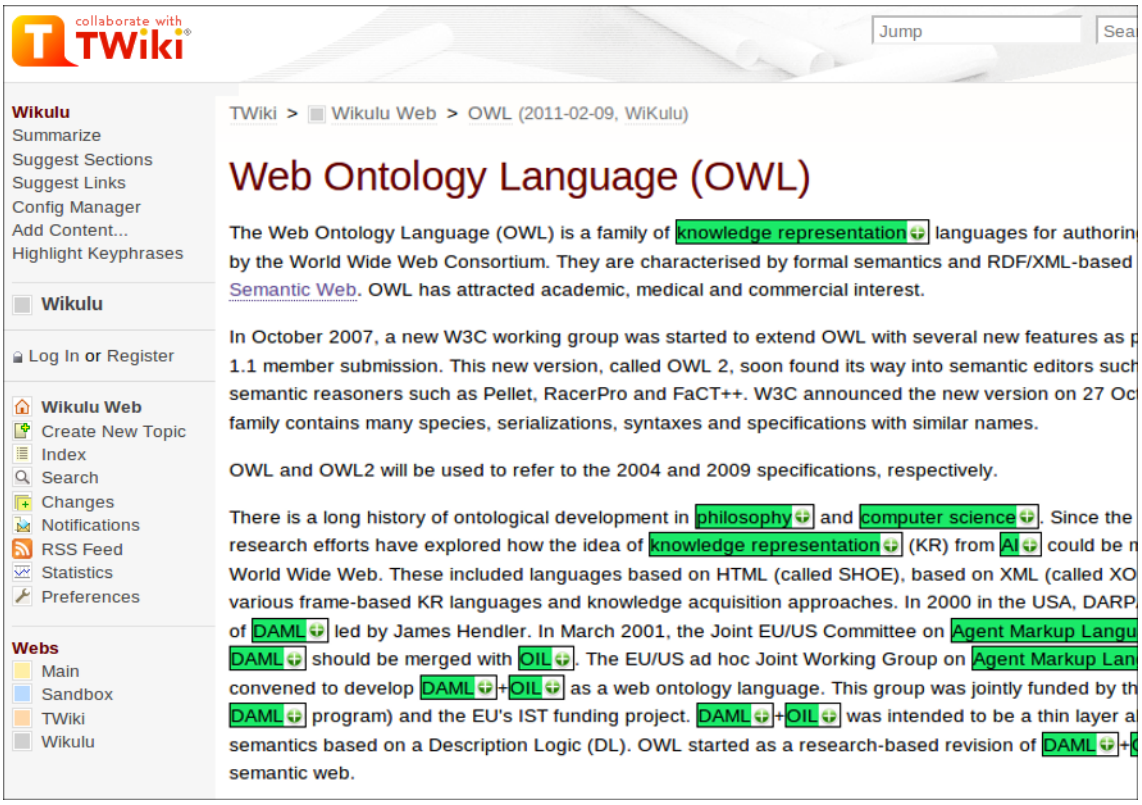


Figure 6.3: Automatic discovery of links to other wiki articles. Suitable text phrases to place a link on are highlighted in green.

TWiki.[4] The additional user interface components are integrated into the default toolbar (highlighted by a red box in the screenshot). In the first example (see Figure 6.2), the user has requested keyphrase highlighting in order to quickly get an idea about the main content of the wiki article. Wikulu then invokes the corresponding NLP component, and highlights the returned keyphrases in the article. In the second example (see Figure 6.3), the user has requested link suggestion, which first identifies relevant concepts in the article and then offers the possibility to manually select links which are then automatically added to the wiki text.

Wikulu builds upon a modular architecture, as depicted in Figure 6.4. It acts as an HTTP proxy server which intercepts the communication between the web browser and the target wiki engine, while it allows to run any *Apache UIMA*-compliant NLP component using an extensible plugin mechanism.

In the remainder of this section, we introduce each module: (a) the proxy server which allows to add Wikulu to any target wiki engine, (b) the JavaScript injection that bridges the gap between the client- and server-side code, (c) the plugin manager which gives access to any *Apache UIMA*-based NLP component, and (d) the wiki abstraction layer which offers a high-level interface to typical wiki operations such as reading and writing the wiki content.

[4] As screenshots only provide a limited overview of Wikulu's capabilities, we refer the reader to a screencast which gives a broader overview: `https://www.youtube.com/watch?v=sPX_D5fy4Fs` (last accessed 2014-09-01)

Proxy Server Wikulu is designed to work with any underlying wiki engine such as *MediaWiki* or *TWiki*. Consequently, we implemented it as an HTTP proxy server which allows it to be enabled at any time by changing the proxy settings of a user's web browser. The proxy server intercepts all requests between the user who interacts with her/his web browser, and the underlying wiki engine. For example, Wikulu passes certain requests to its language processing components, or augments the default wiki toolbar by additional commands. We elaborate on the latter in the following paragraph.

JavaScript Injection Wikulu modifies the requests between web browser and target wiki by injecting custom client-side JavaScript code. Wikulu is thus capable of altering the default behavior of the wiki engine, e.g. replacing a keyword-based retrieval by enhanced search methods, adding novel behavior such as additional toolbar buttons or advanced input fields, or augmenting the originating web page after a certain request has been processed, e.g. an NLP algorithm has been run.

Plugin Manager Wikulu does not perform language processing itself. It relies on *Apache UIMA*-compliant NLP components which use wiki pages (or parts thereof) as input texts. Wikulu offers a sophisticated plugin manager which takes care of dynamically loading those NLP components. The plugin loader is designed to run plugins either every time a wiki page loads, or manually by picking them from the augmented wiki toolbar.

The NLP components are available as server-side Java classes. Via direct web remoting[5], those components are made accessible through a JavaScript proxy object. Wikulu offers a generic language processing plugin which takes the current page contents as input text, runs an NLP component, and writes its output back to the wiki. To run a custom *Apache UIMA*-compliant NLP component with Wikulu, one just needs to plug that particular NLP component into the generic plugin. No further adaptations to the generic plugin are necessary. However, more advanced users may create fully customized plugins.

Wiki Abstraction Layer Wikulu communicates with the underlying wiki engine via an abstraction layer. That layer provides a generic interface for accessing and manipulating the underlying wiki engine. Thereby, Wikulu can both be tightly coupled to a certain wiki instance such as *MediaWiki* or *TWiki*, while being flexible at the same time to adapt to a changing environment. New adaptors for other target wiki engines such as *Confluence*[6] can be added with minimal effort.

As a summary, Wikulu is an extensible user interface which integrates automatic text structuring with wikis. Due to its modular and flexible architecture, we envision that Wikulu can support wiki users in small focused closed domain environments as well as in large-scale communities such as Wikipedia.

[5] `http://directwebremoting.org` (last accessed: 2014-12-07)

[6] `http://www.atlassian.com/software/confluence` (last accessed: 2014-12-07)

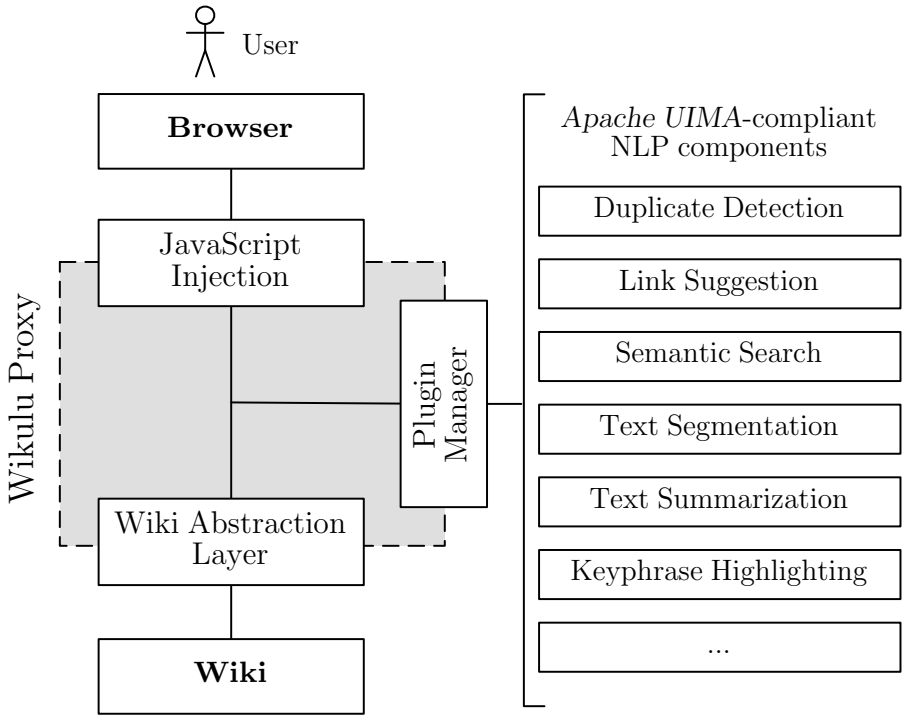


Figure 6.4: Wikulu acts as a proxy server which intercepts the communication between the web browser and the underlying wiki engine. Its plugin manager allows to integrate any *Apache UIMA*-compliant NLP component.

6.2 open window

In this section we present a user interface capable of structuring texts by enriching it with links to Wikipedia and DBpedia. Wikipedia[7] has become a major source of information across many domains (Ladd, 2009). Although some articles may be biased and lack quality (Ferschke et al., 2013; Flekova et al., 2014), its size and continuously updating nature makes Wikipedia a useful alternative to encyclopedias (Giles, 2005). DBpedia[8] constitutes a structured, machine-readable knowledge base that has been extracted from the Wikipedia document corpus, with the goal to make the knowledge processable automatically. DBpedia describes metadata about senses contained in the Wikipedia corpus in a structured way. When linking text snippets to associated DBpedia database, applications can extract valuable background information from the DBpedia corpus and display important information in a context-sensitive manner.

Linking the information with the text is extremely important (Bizer et al., 2008), but it is not possible to do this manually for large corpora. Digital libraries, news articles, Wikipedia itself, are all examples of collections containing documents that can be linked to Wikipedia articles, making it easier for the users to find more encyclopedic information about a topic of interest. This is similar to DBpedia Spotlight by Mendes et al. (2011).

In the course of an industry cooperation with IMC[9], a German company for e-learning technologies, we integrated our approaches to link identification as a central component

[7] http://www.wikipedia.org
[8] http://dbpedia.org/About
[9] `http://www.im-c.de` (last accessed: 2014-12-07)

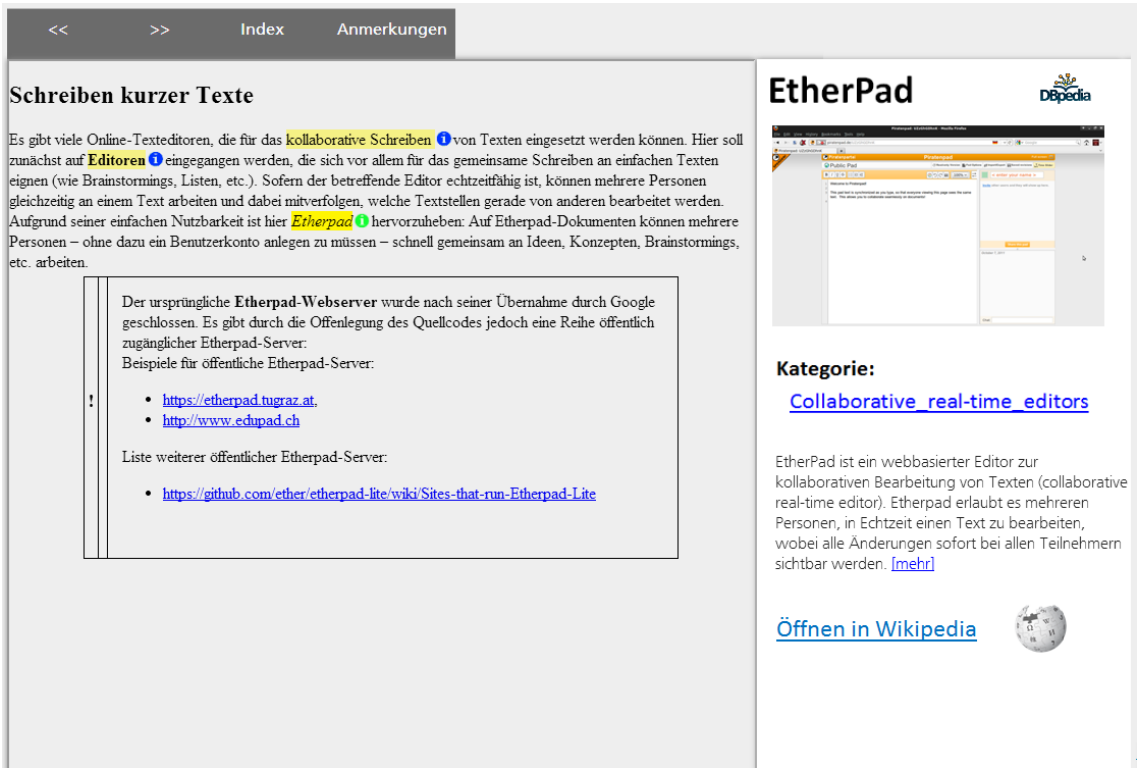


Figure 6.5: Prototype of the system for mention identification with background information for the term *EtherPad*.

of an open, web-based e-learning platform called *Social Navigator*. Designed as a platform to convey knowledge about Social Web skills in vocational education, the Social Navigator provides access to expert information that enables teachers to integrate social media education into the teaching process. It offers training material and useful links about social web tools to trainees. A large set of these learning objects is provided in the form of eBooks, published in the standardized ePUB3 format[10].

As a proof-of-concept, we integrated our approach into the web-based ePUB library *epub.js*[11], which is used in the Social Navigator to render eBooks. We modified the library in a way that, before rendering the content of an eBook page, the text content of the page is passed to the mention identification system. The latter identifies links towards Wikipedia resources. The page content is then modified by adding small "info" symbols to the discovered resources, which indicate that additional information is available.

The user may now click on this *info* symbol to obtain additional information in an info bar at the right of the main content (see Figure 6.5). The additional information is obtained from the DBpedia database. Additional information is automatically extracted from DBpedia resources. Technically, this is implemented by sending a set of parameterized queries—instantiating the DBpedia resource of the entity under consideration—to the public DBpedia SPARQL endpoint at `http://dbpedia.org/sparql` (last accessed: 2014-12-07). Query results are then used to instantiate an HTML5 template, providing the user with structured information about the resource, including information

[10] http://idpf.org/epub/30
[11] http://fchasen.github.io/epub.js/

such as a label, description, image, and links to the Wikipedia pages. The information that was extracted from DBpedia (cf. Figure 6.5) for the term EtherPad consists of a screenshot illustrating the EtherPad software, the category (represented as a link to a survey page listing collaborative real-time editors), an abstract, as well as a link to the corresponding Wikipedia page.

As a summary, open window is a user interface for education purposes. It provides a technique for text structuring by adding links to eBooks and thus helping students to learn about new topics. In the context of this thesis, it seamlessly integrates techniques of text structuring for end users, providing evaluation facilities to further analyze the usefulness of text structuring in the educational setting.

6.3 Chapter Summary

The main contributions of this chapter can be summarized as follows:

Contribution 6.1: *In collaboration with other researchers, we created Wikulu, a wiki proxy for seamlessly integrating natural language processing components into any wiki-based knowledge management system.*

Contribution 6.2: *In the context of a Software Campus project, we presented open window, an online system for automatically linking educational content with information from Wikipedia and DBpedia.*

In the final chapter of the thesis, we presented two prototypes for text structuring systems. Both prototypes relate text structuring techniques to a user and allow for evaluating the usefulness of approaches to text structuring in user studies. They also serve as a demonstration system for potential corporate users.

Wikulu is a proxy for wiki engines and especially useful for organizations with an existing wiki. The existing content is not changed directly, instead Wikulu supports users to interact with the wiki more efficiently. Showing keyphrases for an article helps readers to faster understand the content of an article. Adding links to an article in the wiki does not require knowledge about all other articles, but Wikulu suggests which phrase could be linked to which other article in the wiki. Wikulu is constructed as a modular system, which allows for integrating further text structuring techniques.

The second prototype, open window, is an extension to a learning management system, which links relevant terms in a document to entities in DBpedia and Wikipedia. This helps learners to independently organize their learning activities in self-directed learning. With more and more educational content available online, this offers the possibility to extend course-specific content with additional information, which is available in Wikipedia.

Chapter 7

Conclusions

In this chapter, we will summarize the contributions of this thesis and present suggestions for future research. Figure 7.1 gives an overview of the thesis' contents, including the three text structuring techniques.

7.1 Summary

In this thesis, we described and evaluated techniques for automatic text structuring. We showed the usefulness of text structuring techniques when working with textual data. Readers are likely to understand structured documents faster than documents lacking any kind of structure. Keyphrases give insights into the document's topics, a table-of-contents helps finding the relevant segments in a document, and links enable users to inspect related documents.

We conducted a user survey among 88 Internet users. They answered questions related to the environments and tasks that they solve using the Internet. Almost all (87) of the participants use the Internet at home and at work but also while traveling, or waiting. Communication, searching for information, and social media usage are among the most frequent user tasks. We asked participants to rate techniques according to their usefulness for text structuring and found that keyphrases, table-of-contents, and links are among the highest rated techniques.

7.1.1 Scenarios

We defined three common user scenarios based on the highest ranked tasks of Internet users. These scenarios reflect frequent situations: the *focused searcher* is interested in information for a single broader topic, the *recreational news reader* is interested in news in a specific or general domain, and the *knowledge worker in a company* looks for very specific information, which is typically found in a corporate Intranet.

For some of the text structuring techniques, there already exist approaches. We showed, however, that these approaches do not work equally well in every scenario. For some scenarios, one approach yields better results, while it fails in another. We evaluate approaches to text structuring with several datasets reflecting the selected scenarios. Additionally, we developed new approaches to text structuring to improve state of the art results.

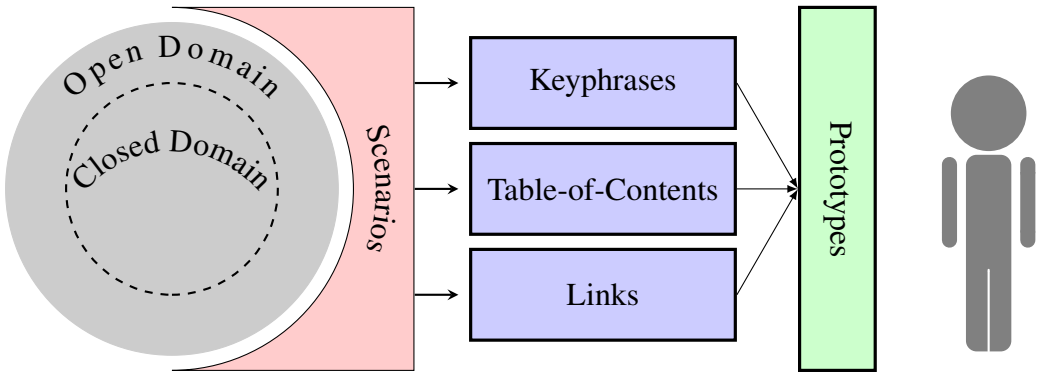


Figure 7.1: Graphical overview of the contents which are covered in this thesis.

7.1.2 Approaches to automatic text structuring

In the following, we give a summary of our findings for each of the investigated techniques.

Keyphrase identification We analyzed approaches to keyphrase extraction and assignment. Extraction approaches can only identify keyphrases appearing in the document text and assignment approaches are limited to keyphrases from a predefined set. For keyphrase extraction, we rely on unsupervised approaches. In addition to state of the art approaches, we evaluate a filtering approach and an extension with decompounding for German. The filtering approach is based on a list of frequently used keyphrases, which professional indexers created. It yields a higher precision. Decompounding is the process of splitting compounds into meaningful parts. Compounds are frequently used in germanic languages and are a challenge for natural language processing. We integrated the ASV toolbox (Biemann et al., 2008) and compared it to other state of the art approaches.

Our evaluation shows that keyphrase extraction approaches yield stable results for our datasets. The state of the art approach TextRank is outperformed by at least one of the tf-idf configurations. Keyphrase assignment and keyphrase filtering yield very good results if there is a predefined set of keyphrases. Our assignment approach using multi-label classification yields high precision but limited recall. Selecting a smaller label set further increases precision while reducing recall. Using a controlled vocabulary for filtering keyphrases, yields higher precision and higher recall. However, this requires a set of previously collected frequent keyphrases. Such sets might exist in corporate environments, where companies hire expert annotators to enforce a better knowledge management. With the German decompounding extension for keyphrase extraction, results improve for shorter documents as more accurate frequencies can be gathered. Results for the decompounding extension decrease for the tf-idf$_{web}$ configuration, because additional non-word keyphrases are created and ranked high (because of their low document frequency).

In summary, we have presented a wide range of approaches to keyphrase identification, including extraction and assignment approaches. Our results show the strengths and weaknesses of each approach. We conclude that there is not a single approach suitable for every scenario but one is sometimes better suited than another. We analyzed these approach-specific characteristics and described their suitability for the different scenar-

ios.

Table-of-contents generation A table-of-contents provides a structured overview of a document's contents. The task of table-of-contents generation is threefold: (i) segmenting the text, (ii) generating titles for segments, and (iii) structuring the segments hierarchically. Almost all documents on the Internet are already segmented, thus we presented solutions for the remaining two subtasks.

For segment title generation, we presented unsupervised and supervised approaches. The unsupervised approaches perform best on two of the three datasets (best on Gutenberg and Cormen), while the supervised system using character n-grams yields best results on the Wikipedia dataset. This is due to many reoccurring titles in Wikipedia articles. In the other two datasets, titles reoccur less frequently which results in a poor accuracy.

For hierarchy identification, we present a supervised system using n-grams, shared entities, noun chunk overlap, keyphrases, and word frequency information as features. Our supervised approaches yield an accuracy of up to 91% on the Gutenberg dataset (87% and 86% on the other datasets) for classifying whether a segment is higher, lower, or on the same level as the previous segment. A manual inspection of resulting table-of-contents has shown that they are similar to the original table-of-contents. A hierarchically structured table-of-contents helps users in all scenarios to quickly get an overview of the document's topics and decide which segment is most promising to read.

Link identification Link identification integrates the tasks of identifying mentions in a text and identifying entities for the mentions to create links from mentions to entities. For mention identification, we used a randomly selected subset of Wikipedia articles and used the existing links in the articles as a gold standard. Link-based (using the existing Wikipedia link structure) and title-based (using Wikipedia titles) approaches outperform text-based approaches by a wide margin. When reducing the number of links in the training data, however, text-based approaches outperform link-based approaches. This is also true for the entity linking task, for which no title-based approaches exist. Hence, link-based approaches are best suited for linking to Wikipedia, text-based approaches are better suited for closed domain environments or to identify links inside a document collection. We presented results for entity linking with the Personalized PageRank algorithm, which uses the Wikipedia link structure.

We further investigate the related task of computing the similarity of words, which makes use of link information. Most measures compute similarity on a word-level, making it impossible to differentiate the value of similarity, e.g. to the word *bass* in only one of the senses *fish* or *instrument*. We propose a sense-level similarity measure, considering senses from a sense inventory. With this measure, we are able to decide which sense of an ambiguous word has a higher similarity to another sense with 70% accuracy. The sense-level similarity measure outperforms existing word similarity measures on a re-rated version on the Rubenstein Goodenough dataset (Rubenstein and Goodenough, 1965). It performs even better when the annotators are aware of the possible senses in the sense inventory used by the similarity measure.

7.1.3 Prototypes for text structuring systems

In addition to the three text structuring techniques (keyphrases, table-of-contents, and links), we presented prototypes for text structuring systems. These prototypes integrate the presented text structuring techniques to support users in their tasks. They are also the base for user studies (Schwarz et al., 2010) and demonstrators for the usefulness of the implemented approaches.

We presented *Wikulu* and *open window*, two prototypes integrating natural language processing components. With Wikulu, we provide a flexible framework for testing components in a wiki scenario, helping users to add, organize, and find information. This is especially useful in a corporate wiki which often lacks structure (Buffa, 2006). With open window, we integrated our component for link identification in an e-learning scenario. The automatic link identification component helps students by adding links to Wikipedia and DBpedia in eBooks.

The prototypes demonstrate the applicability of automatic approaches and we believe that the approaches presented in this thesis will be used as part of text structuring systems—especially in corporate environments—in the future. Additionally, the prototypes and the developed approaches enable researchers to further improve automatic text structuring and conduct experiments in related disciplines.

7.2 Future Research Directions

In this thesis, we mainly focused on unsupervised approaches because they do not require any training data and are thus in principle domain independent. For future work, we propose to investigate to what extent supervised approaches improve results on our test datasets (using a cross-validation evaluation setup). We also propose to evaluate if supervised approaches improve results when trained on data from another domain. We believe that results are influenced by topics (e.g. fiction and non-fiction literature) and genres (e.g. newspaper text and social media). Related work has shown that using domain data improves results (Ferrucci et al., 2010) but it is still unclear how much domain data is necessary. Using some domain data to enrich a large portion of domain independent data has been shown as an alternative to creating language models (Durme and Osborne, 2014) and can be beneficial for approaches to text structuring.

7.2.1 Approaches to automatic text structuring

We further see future research directions for improving the presented approaches to text structuring on an algorithmic level. We will now report on possible improvements for each of the text structuring techniques.

Keyphrase identification Our experiments have shown that results for unsupervised approaches highly depend on the domain of the document. Even different configurations of the tf-idf approach do not yield stable results across all datasets. Combining different configurations in a supervised approach might lead to better results across all datasets. Improvements of decompounding approaches will further result in an improvement of keyphrase extraction approaches when compounds can be identified more accurately. This

also holds for any improvements leading to more accurate counting of word frequencies, e.g. detecting hyphenation, lemmatization, and acronym expansion. Especially when processing user-generated data, preprocessing targeted towards possibly ungrammatical text might improve results for keyphrase identification.

There are further possibilities for improvement by combining extraction and assignment approaches. A linear combination of extraction and assignment approaches has already shown to improve results (Erbs et al., 2013a) but there is still potential for improvement by considering scores for keyphrases from extraction and assignment approaches. Our experiments with a controlled vocabulary have shown to improve results but it requires manual effort to create such a controlled vocabulary. For future work, the automatic population of topic words, which represent good keyphrases, is a potential alternative to manually creating a controlled vocabulary.

Table-of-contents generation To improve our supervised hierarchy identification system, we can extend our feature set with additional features based on similarity metrics. A high similarity of two document segments can be evidence for the same hierarchy level as both segments share the same topic. Further features based on topic modeling have a similar effect. In our evaluation scenario, we evaluated table-of-contents generation by counting how often the level difference of segment pairs is correctly identified. We propose to evaluate all table-of-contents in a user setting and use a Turing test (Turing, 1950). This allows for evaluating a table-of-contents as an ensemble of hierarchies.

For identifying segment titles, we can rely on improvements in keyphrase extraction and assignment. Especially, creating a controlled vocabulary with potential segment titles may improve results. We propose to analyze if segment titles follow certain patterns. Identifying such patterns will help reducing the set of title candidates.

Link identification For link identification, we propose to develop further techniques for the global linking of entities (all mentions in a document collectively). Identifying links for a large set of mentions is merely a computational issue. Optimization techniques can be applied to efficiently compute edge weights between two possible entities. To link entities to other knowledge bases without any existing link structure, first unsupervised mention identification and entity link approaches need to be applied. The resulting links can then be curated manually and further used for supervised link identification. For mention identification, unsupervised keyphrase extraction approaches, and for entity linking, approaches from information retrieval can be applied.

7.2.2 Future research in related disciplines

In this thesis, we have shown that one needs to carefully select an approach based on the scenario. Approaches do not perform equally well on every evaluation dataset and in every scenario. However, many novel algorithms, which are presented nowadays, improve the state of the art just on standard benchmark datasets. We believe that in many cases, the reported improvement is due to a better tuning to the dataset and not due to a more general and better approach. This raises questions about the applicability of approaches to user problems. Wagstaff (2012) focuses on machine learning research and asks: “What’s it good for?” In her opinion, “machine learning research has lost its connection to problems

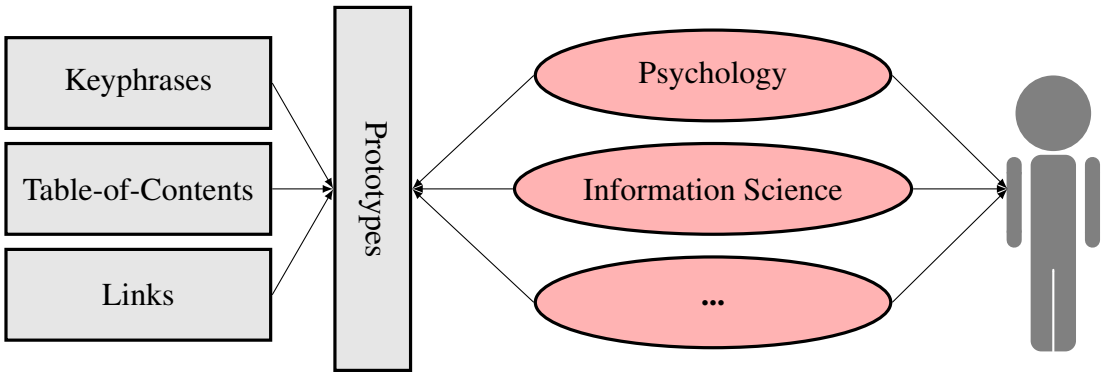


Figure 7.2: Future research direction of related disciplines.

of importance to the larger world of science and society". Novel approaches need to show their impact on society or on scientific disciplines beyond a single benchmark dataset (Rudin and Wagstaff, 2013).

Measuring the benefit of approaches for users is a very difficult task as it needs to be isolated from external factors. The design of the user interface, the academic background of the user, and the task at hand may have an impact on the results. Controlling for all these external factors in user studies is out of the scope of this thesis. However, we have laid foundations for future studies of text structuring approaches. To foster future research on automatic text structuring, we propose a collaboration of computational linguistics with related disciplines. Figure 7.2 shows the role of related disciplines regarding the usage of text structuring techniques from the perspective of the user via a prototype. We now describe possible future research directions for these disciplines.

Psychology We have shown that our approaches for automatic text structuring improve results on evaluation datasets, but from a psychological point of view, the research question is, whether these approaches support users in their tasks. Schwarz (2010) assumes that automatic text structuring reduces the *cognitive load* (Sweller, 1988) and thus enables users to concentrate on the task itself. A learner, who wants to get informed about a topic, should focus on the information and not on finding articles about terms in the text (which can be done automatically with link identification).

Psychologists' expertise is further required to conduct user studies which are controlled for external influences. A user without any computer experience requires a different kind of automatic text structuring support than a user with many years of computer experience. The complexity of a user interface influences how a user interacts with text structuring techniques. So far, in evaluation settings with humans, it is hard to distinguish the effect of better approaches and better prototypes to ensure the validity of psychological tests (Cronbach, Lee J and Meehl, 1955; Borsboom et al., 2003).

Information science In this thesis, we have defined scenarios in which automatic text structuring helps users. We have not focused on very specific user groups in isolation.[1] In information science, researchers investigate the question how a very specific group of users search for information, e.g. lawyers (Kuhlthau and Tama, 2001), engineers (Robin-

[1] We only differentiated between different environments, e.g. at home and at work.

son, 2010), or librarians (Brown and Ortega, 2005). Any differences of their search behavior affects their usage of automatic text structuring techniques and should be further investigated.

Throughout this thesis, we did not specify different types of information. However, in information science, Kuhlthau (1993) distinguishes between general background information, faceted background information, and specific information. Vakkari (2000) analyzes in which stage of a writing task students seek for which types of information. We believe that depending on the type of information desired, some approaches to text structuring may be better suited than others.

Other disciplines Based on the result of this thesis, there are numerous further research directions in related disciplines. Different techniques for text structuring enrich a text with additional information (links mark phrases in a document, while there is only one table-of-contents for a document). This has implications for the design of the user interface, which may include different kinds of additional information. A challenge for the design of the user interface is to include this meta information without overloading the interface (Norman, 2002).

In business informatics, researchers deal with licensing models of software products (Harnisch and Knaf, 2014). Using approaches to automatic text structuring in a corporate environment may lead to a higher productivity and thus is a valuable asset for corporations. Developers of text structuring systems have multiple distribution channels, including offering it as software-as-a-service, or by offering service contracts. It is an open question, which channel is best suited for text structuring systems.

In cultural studies, researchers investigate the effect of technology on society from an anthropological perspective (Pfaffenberger, 1992; Avison and Myers, 1995; Hughes, 2004). Text structuring techniques support users by automating tasks, which were previously performed manually. We assume that users endorse any technological support, but users may also perceive automatic systems as a threat to their job or a limitation of their freedom. It would also be interesting to investigate the long-term implications of using automatic approaches. Are users still able to deal with unstructured text and does it change the way they search?

7.3 Closing Remarks

The Internet has diversified and raised the sources of information considerably. Not only are there news agencies but also social media channels like Twitter[2] and personal websites. They all provide very different forms of information. However, a large number of the documents providing information lacks any kind of structure, which makes it hard for users to quickly retrieve information. However, the content may still contain valuable information for a user and automatic text structuring enables users to make better use of it. Keyphrases, table-of-contents, and links are useful techniques to better understand the content of documents and thus help users to faster fulfill their information need. The amount of unstructured information in blogs or forums is further increasing. Therefore, automatic text structuring, the core of this thesis, is of considerable importance in years to come.

[2] `https://twitter.com/` (last accessed: 2014-12-18)

We analyzed approaches to text structuring. However, we have shown that there is not a single approach suitable in all scenarios. By using training data, we can improve performance on specific datasets, where training data is not always available. We thus analyzed characteristics of approaches to automatic text structuring and described the best approaches for every scenario. We believe that our findings will serve as a foundation for future text structuring systems, which can be applied both in educational environments and in corporate environments. It will also serve for future interdisciplinary research concerning the role of text structuring in a user's search for information.

Appendix A

Software Packages

A.1 DKPro Keyphrases

A.1.1 Introduction

Keyphrases are single words or phrases that provide a summary of a text (Tucker and Whittaker, 2009) and thus might improve searching (Song et al., 2006) in a large collection of texts. As manual extraction of keyphrases is a tedious task, a wide variety of keyphrase extraction approaches has been proposed. Only few of these are freely available which makes it hard for researchers to replicate previous results or use keyphrase extraction in some other application, such as information retrieval (Manning et al., 2008), or question answering (Kwok et al., 2001).

In this section, we describe DKPro Keyphrases, our framework for keyphrase extraction. It integrates a wide range of state of the art approaches to keyphrase extraction that can be directly used with limited knowledge of programming. However, for developers of new keyphrase extraction approaches, DKPro Keyphrases also offers a programming framework for developing new extraction algorithms and for evaluation of resulting effects. DKPro Keyphrases is based on the Unstructured Information Management Architecture (Ferrucci and Lally, 2004), which provides a rich source of libraries with preprocessing components.

A.1.2 Architecture

The architecture of DKPro Keyphrases models the five fundamental steps of keyphrase extraction: (i) Reading of input data and enriching it with standard linguistic preprocessing, (ii) selecting phrases as keyphrase candidates based on the preprocessed text, (iii) filtering selected keyphrases, (iv) ranking remaining keyphrases, and (v) evaluating ranked keyphrases against a gold standard. This process is visualized in Figure A.1. In this section, we will describe details of each step, including components already included in DKPro Keyphrases.

A.1.3 Preprocessing

DKPro Keyphrases relies on UIMA-based preprocessing components developed in the natural language processing framework DKPro Core (Eckart de Castilho and Gurevych,

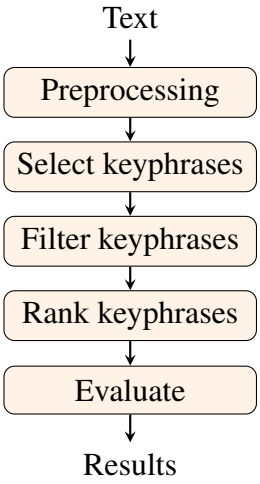


Figure A.1: Architecture overview of DKPro Keyphrases.

2014; Gurevych et al., 2007a; Eckart de Castilho and Gurevych, 2009). Thus, a wide range of linguistic preprocessing components are readily available such as word segmentation, lemmatization, part-of-speech tagging, named entity recognition, syntactic parsing, or co-reference resolution.

A.1.4 Selecting Keyphrases

In this step, DKPro Keyphrases selects all phrases as keyphrases that match user-specified criteria. A criterium is typically a linguistic type, e.g. tokens, or more sophisticated types such as noun phrases. The resulting list of keyphrases should cover all gold keyphrases and at the same time be as selective as possible. We use the following sentence with the two gold keyphrases "dog" and "old cat" as a walk-through example:

> A [dog] chases an [old cat] in my garden.

Taking all uni- and bi-grams as keyphrases will easily match both gold keyphrases, but it will also result in many other less useful keyphrases like "in my".

In the given example, the keyphrase list consists of nine tokens (lemmas, resp.) but covers only one gold keyphrase (i.e. "dog"). Noun chunks and named entities are alternative keyphrases, limiting the set of keyphrases further. Experiments where noun chunks are selected as keyphrases perform best for this example. Named entities are too restrictive, but applicable for identifying relevant entities in a text. This is useful for tasks that are targeted towards entities, e.g. for finding experts (Dörner et al., 2007) in a collection of domain-dependent texts. The selection of a linguistic type is not limited, as preprocessing components might introduce further types.

A.1.5 Filtering

Filtering can be used together with over-generating selection approaches like taking all n-grams to decrease the number of keyphrases before ranking. One possible approach is based on POS patterns. For example, using the POS patterns, `Adjective-Noun`, `Adjective`, and `Noun` limits the set of possible keyphrases to "dog", "old cat", "cat",

and "garden" in the previous example. This step can also performed as part of the selection step, however, keeping it separated enables researchers to apply filters to keyphrases of any linguistic type. DKPro Keyphrases provides the possibility to use controlled-vocabulary keyphrase extraction by filtering out all keyphrases which are not included in a keyphrase list.

Developers of keyphrase extraction approaches can create their own filter simply by extending from a base class and adding filter-specific code. Additionally, DKPro Keyphrases does not impose workflow-specific requirements, such as a fixed number of filters. This leaves room for keyphrase extraction experiments testing new or extended filters.

A.1.6 Ranking

In this step, a ranker assigns a score to each remaining keyphrase candidate. DKPro Keyphrases contains rankers based on the candidate position, frequency, tf-idf, TextRank (Mihalcea and Tarau, 2004b), and LexRank (Erkan and Radev, 2004).

DKPro Keyphrases also contains a special extension of tf-idf, called tf-idf$_{web}$, for which Google Web1T (Brants and Franz, 2006) is used for obtaining df counts from a large corpus. In case of keyphrase extraction for a single document or for domain-independent keyphrase extraction, Web1T provides reliable n-gram statistics without having any domain-dependence.

A.1.7 Evaluation

DKPro Keyphrases ships with all the metrics that have been traditionally used for evaluating keyphrase extraction. Kim et al. (2010) use precision and recall for a different number of keyphrases (5, 10 and 15 keyphrases). These metrics are widely used for evaluation in information retrieval. Precision@5 is the ratio of true positives in the set of extracted keyphrases when 5 keyphrases are extracted. Recall@5 is the ratio of true positives in the set of gold keyphrases when 5 keyphrases are extracted. Moreover, DKPro Keyphrases evaluates with MAP and R-precision. MAP is the mean average precision of extracted keyphrases from the highest scored keyphrase to the total number of extracted keyphrases. For each position in the rank, the precision at that position will be computed. Summing up the precision at each recall point and then taking its average will return the average precision for the text being evaluated. The mean average precision will be the mean from the sum of each text's average precision from the dataset. R-precision is the ratio of true positives in the set of extracted keyphrases, when the set is limited to the same size as the set of gold keyphrases (Zesch and Gurevych, 2009).

A.1.8 Experimental framework

In this section, we show how researchers can perform experiments covering many different configurations for preprocessing, selection, and ranking. To facilitate the construction of experiments, the framework contains a module to make its architecture compatible to the DKPro Lab framework (Eckart de Castilho and Gurevych, 2011), thus allowing to sweep through the *parameter space* of configurations. The parameter space is the combination of all possible parameters, e.g. one parameter with two possible values for

preprocessing and a second parameter with two values for rankers lead to four possible combinations. We refer to *parameter sweeping experiments* when running the experiment with all possible combinations.

DKPro Keyphrases divides the experimental setup in three *tasks*. Tasks are processing steps defined in the Lab framework, which – in case of keyphrase extraction – are based on the steps described in Section A.1.2. In the first task, the input text is fed into a pipeline and preprocessed. In the second task, the keyphrases are selected and filtered. In the third and final task, they are ranked and evaluated. The output of the first two tasks are serialized objects which can be processed further by the following task. The output of the third task is a report containing all configurations and results in terms of all evaluation metrics.

The division into three tasks speeds up processing of the entire experiment. Each task has multiple configuration parameters which influence the forthcoming tasks. Instead of running the preprocessing tasks for every single possible combination, the intermediate objects are stored once and then used for every possible configuration in the keyphrase selection step.

To illustrate the advantages of experimental settings in DKPro Keyphrases, we run the previously used example sentence through the entire parameter space. Hence, tokens, lemmas, n-grams, noun chunks, and named entities will be combined with all filters and all rankers (not yet considering all possible parameters). This results in more than 10,000 configurations. Although the number of configurations is high, the computation time is low[1] as not the entire pipeline needs to run that often. This scales well for longer texts.

The experimental framework runs all possible combinations automatically and collects individual results in a report, such as a spreadsheet or text file. This allows for comparing results of different rankers, mitigating the influence of different preprocessing and filtering components. This way, the optimal experimental configuration can be found empirically. It is a great improvement for researchers because a variety of system configurations can be compared without the effort of reimplementing the entire pipeline.

Code example A.1 shows the main method of an example experiment, selecting all tokens as possible keyphrases and ranking them with their tf-idf values. Lines 1 to 34 show values for dimensions which span the parameter space. A dimension consists of an identifier, followed by one or more values. Lines 36 to 40 show the creation of tasks, and in lines 42 to 48 the tasks and a report are added to one *batch task*, which is then executed. Researchers can run multiple configurations by setting multiple values to a dimension. Line 25 shows an example of a dimension with two values (using the logarithm or unchanged text frequency), in this case two configurations[2] for the ranker based on tf-idf scores.

Listing A.1: Example experiment

```
1 ParameterSpace params = new ParameterSpace(
2  Dimension.create("language", "en"),
3  Dimension.create("frequencies", "web1t"),
4  Dimension.create("tfidfFeaturePath", Token.class"),
5  Dimension.create("dataset", datasetPath),
6  Dimension.create("goldSuffix", ".key"),
```

[1] Less than five minutes on a desktop computer with a 3.4 GHz 8-core processor.

[2] DKPro Keyphrases provides ways to configure experiments using Groovy and JSON.

```
7
8 //Selection
9  Dimension.create("segmenter", OpenNlpSegmenter.class),
10 Dimension.create("keyphraseFeaturePath", Token.class),
11 //PosSequence filter
12 Dimension.create("runPosSequenceFilter", true),
13 Dimension.create("posSequence", standard),
14 //Stopword filter
15 Dimension.create("runStopwordFilter", true),
16 Dimension.create("stopwordlists", "stopwords.txt"),
17 // Ranking
18 Dimension.create("rankerClass", TfidfRanking.class),
19 //TfIdf
20 Dimension.create("weightingModeTf", NORMAL, LOG),
21 Dimension.create("weightingModeIdf", LOG),
22 Dimension.create("tfidfAggregate", MAX),
23 //Evaluator
24 Dimension.create("evalMatchingType", MatchingType.Exact),
25 Dimension.create("evalN", 50),
26 Dimension.create("evalLowercase", true),
27 Dimension.create("evalType", EvaluatorType.Lemma),
28 );
29
30 Task preprocessingTask = new PreprocessingTask();
31 Task filteringTask = new KeyphraseFilteringTask();
32 candidateSelectionTask.addImport( preprocessingTask,
      PreprocessingTask.OUTPUT, KeyphraseFilteringTask.INPUT);
33 Task keyphraseRankingTask = new KeyphraseRankingTask();
34 keyphraseRankingTask.addImport( filteringTask,
      KeyphraseFilteringTask.OUTPUT, KeyphraseRankingTask.INPUT);
35
36 BatchTask batch = new BatchTask();
37 batch.setParameterSpace(params);
38 batch.addTask(preprocessingTask);
39 batch.addTask(candidateSelectionTask);
40 batch.addTask(keyphraseRankingTask);
41 batch.addReport( KeyphraseExtractionReport.class);
42 Lab.getInstance().run(batch);
```

One possible use case for the experimental framework is the evaluation of new preprocessing components. For example, keyphrase extraction should be evaluated with Twitter data: One collects a dataset with tweets and their corresponding keyphrases (possibly, the hash tags). The standard preprocessing will most likely fail as non-canonical language will be hard to process (e.g. *hash tags* or emoticons).

The preprocessing components can be set as a parameter and compared directly without changing the remaining parameters for filters and rankers. This allows researchers to perform extrinsic evaluation of their components in a keyphrase extraction task.

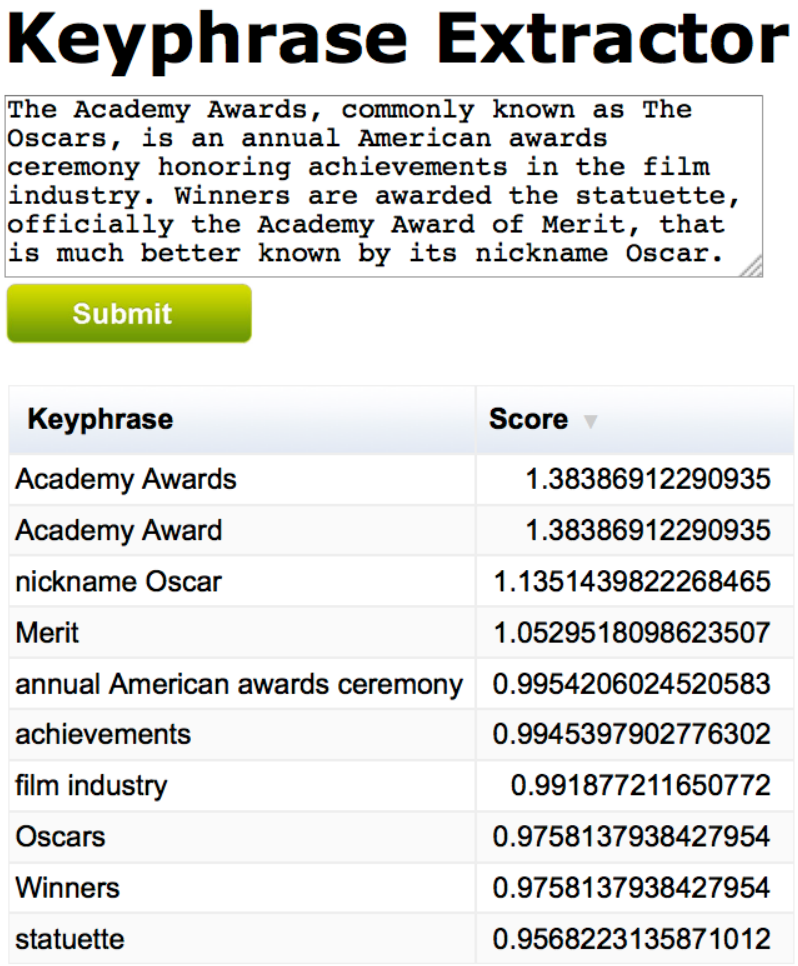


Figure A.2: Screenshot of web demo in DKPro Keyphrases.

A.1.9 Visualization and wrappers

To foster analysis of keyphrase extraction experiments, we created a web-based visualization framework with Spring[3]. It allows for running off-the-shelf experiments and manually inspecting results without the need to install any additional software. Figure A.2 shows a visualization of one pre-configured experiment. The web demo is available online.[4] Currently, a table overview of extracted keyphrases is implemented, but developers can change it to highlighting all keyphrases. The latter is recommended for a binary classification of keyphrases.[5] This is the case, if a system only returns keyphrases with a score above a certain threshold. The table in Figure A.2 shows keyphrases with the assigned scores, which can be sorted to get a ranking of keyphrases. However, the visualization framework does not provide any evaluation capabilities.

To help new users of DKPro Keyphrases, it includes a module with two demo experiments using preconfigured parameter sets. This is especially useful for applying keyphrase extraction in other tasks, e.g. text summarization (Goldstein et al., 2000). Both

[3] `http://projects.spring.io/spring-ws/` (last accessed: 2014-12-07)

[4] `https://dkpro.ukp.informatik.tu-darmstadt.de/DKProKeyphrases` (last accessed: 2014-12-07)

[5] With binary classification an unranked list of keyphrases is returned.

demo experiments are frequently used keyphrase extraction systems. The first one is based on TextRank (Mihalcea and Tarau, 2004b), and the second one is based on the supervised system KEA (Witten et al., 1999). Both configurations do not require any additional installation of software packages.

This module offers setters to configure parameters, e.g. the size of co-occurrence windows in case of the TextRank extractor.

A.1.10 Related work

Most work on keyphrase extraction is not accompanied with free and open software. These tools listed in this section allow users to combine different configurations in respect to preprocessing, keyphrase selection, filtering, and ranking. In the following, we give an overview of software tools for keyphrase extraction.

KEA (Witten et al., 1999) provides a Java API, which offers automatic keyphrase extraction from texts. They provide a supervised approach for keyphrase extraction. For each keyphrase, KEA computes frequency, position, and semantic relatedness as features. Thus, for using KEA, the user needs to provide annotated training data. KEA generates keyphrases from n-grams with length from 1 to 3 tokens. A controlled vocabulary can be used to filter keyphrases. The configuration for keyphrase selection and filtering is limited compared to DKPro Keyphrases, which offers capabilities for changing the entire preprocessing or adding filters.

Maui (Medelyan et al., 2009) enhances KEA by allowing the computation of semantic relatedness of keyphrases. It uses Wikipedia as a thesaurus and computes the keyphraseness of each keyphrase, which is the number of times a candidate was used as keyphrase in the training data (Medelyan et al., 2009).

Although Maui provides training data along with their software, this training data is highly domain-specific. A shortcoming of KEA and Maui is the lack of any evaluation capabilities or the possibility to run parameter sweeping experiments. DKPro Keyphrases provides evaluation tools for automatic testing of many parameter settings.

Besides KEA and Mau, which are Java systems, there are several modules in Python, e.g. topia.termextract[6], which offer capabilities for tokenization, part-of-speech tagging and keyphrase extraction. Keyphrase extraction from topia.termextract is based on noun phrases and ranks these according to their frequencies.

BibClassify[7] is a python module which automatically extracts keywords from a text based on the occurrence of terms in a thesaurus. The ranker is frequency-based like topia.termextract. BibClassify and topia.termextract do not provide evaluation capabilities or parameter sweeping experiments.

Besides these software tools, there are web services for keyphrase extraction. AlchemyAPI[8] offers a web service for keyword extraction. It may return keyphrases encoded in various markup languages. TerMine[9] offers a SOAP service for extracting keyphrases from documents and a web demo. The input must be a String and the extracted terms will be returned as a String. Although web services can be integrated easily due to their

[6] `https://pypi.python.org/pypi/topia.termextract/` (last accessed: 2014-12-07)

[7] `http://invenio-demo.cern.ch/help/admin/bibclassify-admin-guide` (last accessed: 2014-12-07)

[8] `http://www.alchemyapi.com/api/keyword-extraction/` (last accessed: 2014-12-07)

[9] `http://www.nactem.ac.uk/software/termine/` (last accessed: 2014-12-07)

protocol stacks, they are not extensible and their replicability cannot be guaranteed over time.

A.1.11 Summary

In this section we presented DKPro Keyphrases, which is a framework for flexible and reusable keyphrase extraction experiments. This helps researchers to effectively develop new keyphrase extraction components without the need to re-implement state of the art approaches.

The UIMA-based architecture of DKPro Keyphrases allows users to easily evaluate keyphrase extraction configurations. Researchers can integrate keyphrase extraction with different existing linguistic preprocessing components offered by the open-source community, They can evaluate these in terms of all commonly used evaluation metrics.

List of Tables

List of Figures

Bibliography

Agirre, E., Chang, A. X., Jurafsky, D. S., Manning, C. D., Spitkovsky, V. I., and Yeh, E. (2009). Stanford-UBC at TAC-KBP. In *Proceedings of Text Analysis Conference (TAC 2009)*.

Agirre, E. and Edmonds, P. (2006). *Word Sense Disambiguation: Algorithms and Applications*. Springer.

Agirre, E., Fellbaum, C., Marchetti, A., and Toral, A. (2010). SemEval-2010 Task 17 : All-words Word Sense Disambiguation on a Specific Domain. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 123–128.

Agirre, E. and Soroa, A. (2009). Personalizing PageRank for Word Sense Disambiguation. In *Proceedings of 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41.

Alfonseca, E., Bilac, S., and Pharies, S. (2008). German Decompounding in a Difficult Corpus. *Proceedings of Computational Linguistics and Intelligent Text Processing*, pages 128–139.

Allan, J. (1996). Automatic Hypertext Link Typing. In *Proceedings of the the Seventh ACM Conference on Hypertext - HYPERTEXT '96*, pages 42–52.

Assael, H. (2005). A Demographic and Psychographic Profile of Heavy Internet Users and Users by Type of Internet Usage. *Journal of Advertising Research*, 45(01):93–123.

Attali, Y. and Burstein, J. (2006). Automated Essay Scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).

Avison, D. E. and Myers, M. D. (1995). Information Systems and Anthropology: An Anthropological Perspective on IT and Organizational Culture. *Information Technology & People*, 8(3):43–56.

Babbage, C. (1864). *Passages from the Life of a Philosopher*. Longman, Green, Longman, Roberts, & Green.

Baker, P. (2004). Querying Keywords Questions of Difference, Frequency, and Sense in Keywords Analysis. *Journal of English Linguistics*, 32(4):346–359.

Baker, P., Hardie, A., and McEnery, T. (2006). *A Glossary of Corpus Linguistics*. Edinburgh University Press Edinburgh.

Banerjee, S. and Pedersen, T. (2002). An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet. In *Computational Linguistics and Intelligent Text*, pages 136–145.

Bär, D., Erbs, N., Zesch, T., and Gurevych, I. (2011a). Wikulu: An Extensible

Architecture for Integrating Natural Language Processing Techniques with Wikis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. System Demonstrations*, pages 74–79.

Bär, D., Zesch, T., and Gurevych, I. (2011b). A Reflective View on Text Similarity. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 515–520, Hissar, Bulgaria.

Bär, D., Zesch, T., and Gurevych, I. (2013). DKPro Similarity: An Open Source Framework for Text Similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*, pages 121–126, Sofia, Bulgaria.

Barker, K. and Cornacchia, N. (2000). Using Noun Phrase Heads to Extract Document Keyphrases. *Advances in Artificial Intelligence*, 1882:40–52.

Barnett, J. E. (2003). Do Instructor-provided Online Notes Facilitate Student Learning? Jerrold E. Barnett. *The Journal of Interactive Online Learning*, 2(2):1–7.

Baron, L., Tague-Sutcliffe, J., Kinnucan, M. T., and Carey, T. (1996). Labeled, Typed Links As Cues when Reading Hypertext Documents. *Journal of American Society of Information Science*, 47(12):896–908.

Baroni, M., Matiasek, J., and Trost, H. (2001). Predicting the Components of German Nominal Compounds. In *Proceedings of the European Conference on Artificial Intelligence*, pages 1–12.

Barr, C., Jones, R., and Regelson, M. (2008). The Linguistic Structure of English Web-search Queries. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1021–1030.

Bartram, L., Ho, A., Dill, J., and Henigman, F. (1995). The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces. In *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology*, pages 207–215.

Barzilay, R. and Elhadad, M. (1997). Using Lexical Chains for Text Summarization. In *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

Bauer, L. (1983). *English Word-formation*. Cambridge University Press.

Beleites, C., Neugebauer, U., Bocklitz, T., Krafft, C., and Popp, J. (2013). Sample Size Planning for Classification Models. *Analytica Chimica Acta*, 760(6):25–33.

Biemann, C., Quasthoff, U., Heyer, G., and Holz, F. (2008). ASV Toolbox: A Modular Collection of Language Exploration Tools. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 1760–1767.

Bizer, C., Heath, T., Idehen, K., and Berners-Lee, T. (2008). Linked Data on the Web. In *Proceedings of the 17th International Conference on World Wide Web*, pages 1265–1266.

Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Borsboom, D., Mellenbergh, G. J., and van Heerden, J. (2003). The Theoretical Status of

Latent Variables. *Psychological Review*, 110(2):203–219.

Branavan, S., Deshpande, P., and Barzilay, R. (2007). Generating a Table-of-Contents. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 544–551.

Brandtzaeg, P. B., Heim, J., and Karahasanović, A. (2011). Understanding the New Digital Divide—A Typology of Internet Users in Europe. *International Journal of Human-Computer Studies*, 69(3):123–138.

Brants, T. and Franz, A. (2006). Web 1T 5-Gram Version 1. In *Linguistic Data Consortium*, Philadelphia.

Brody, S. and Diakopoulos, N. (2011). Cooooooooooooooollllllllllllll!!!!!!!!!!!!!!: Using Word Lengthening to Detect Sentiment in Microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 562–570.

Brooke, J. and Hirst, G. (2011). Native Language Detection with 'Cheap' Learner Corpora. In *Twenty Years of Learner Corpus Research. Looking Back, Moving Ahead: Proceedings of the First Learner Corpus Research Conference (LCR 2011)*, pages 37–47.

Brown, C. M. and Ortega, L. (2005). Information-seeking Behavior of Physical Science Librarians : Does Research Inform Practice? *College & Research Libraries*, 66(3):231–247.

Bubenhofer, N. (2009). *Sprachgebrauchsmuster: Korpuslinguistik als Methode der Diskurs- und Kulturanalyse*. Walter de Gruyter.

Buckley, C. and Voorhees, E. M. (2000). Evaluating Evaluation Measure Stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '00*, pages 33–40, New York, New York, USA.

Budanitsky, A. and Hirst, G. (2006). Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47.

Buffa, M. (2006). Intranet Wikis. In *Proceedings of the IntraWebs Workshop 2006 at the 15th International World Wide Web Conference*.

Bughin, J. (2007). How Companies Can Make the Most of User-Generated Content. *McKinsey Quarterly*, (8):1–4.

Bunescu, R. (2006). Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 9–16.

Buzan, T. (1984). *Make the Most of your Mind*. Simon and Schuster.

Calfee, R. C. and Drum, P. (1986). Research on Teaching Reading. *Handbook of Research on Teaching*, 3:804–849.

Cao, Y., Xu, J., Liu, T.-Y., Li, H., Huang, Y., and Hon, H.-W. (2006). Adapting Ranking SVM to Document Retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 186–193.

Capocci, A., Servedio, V. D. P., Colaiori, F., Buriol, L. S., Donato, D., Leonardi, S., and Caldarelli, G. (2006). Preferential Attachment in the Growth of Social Networks: The Internet Encyclopedia Wikipedia. *Physical Review E*, 74:36116.

Carpineto, C., Osiński, S., Romano, G., and Weiss, D. (2009). A Survey of Web Clustering Engines. *ACM Computing Surveys*, 41(3):1–38.

Chang, A., Spitkovsky, V., Yeh, E., Agirre, E., and Manning, C. D. (2010). Stanford-UBC Entity Linking at TAC-KBP. In *Proceedings of Text Analysis Conference (TAC 2010)*.

Chi, E., Gumbrecht, M., and Hong, L. (2007). Visual Foraging of Highlighted Text: An Eye-tracking Study. *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, pages 589–598.

Chiou, C.-C., Lee, L.-T., and Liu, Y.-Q. (2012). Effect of Novak Colorful Concept Map with Digital Teaching Materials on Student Academic Achievement. *Procedia - Social and Behavioral Sciences*, 64(9):192–201.

Choi, F. Y. Y. (2000). Advances in Domain Independent Linear Text Segmentation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, pages 26–33, San Francisco, CA, USA.

Cicognani, A. (2000). Concept Mapping as a Collaborative Tool for Enhanced Online Learning. *Educational Technology & Society*, 3(3):150–158.

Cilibrasi, R. and Vitanyi, P. (2007). The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383.

Cobuild, C. (2006). *Collins COBUILD Advanced Learner's English Dictionary*. Collins Cobuild.

Collins, M. and Roark, B. (2004). Incremental Parsing with the Perceptron Algorithm. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 111–118.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. The MIT Press, 2nd edition.

Cortes, C. and Vapnik, V. (1995). Support-vector Networks. *Machine Learning*, 297:273–297.

Cronbach, Lee J and Meehl, P. E. (1955). Construct Validity in Psychological Tests. *Psychological bulletin*, 52(4):281–302.

Csomai, A. and Mihalcea, R. (2006). Creating a Testbed for the Evaluation of Automatically Generated Back-of-the-book Indexes. In *Proceedings of the Conference on Computational Linguistics and Intelligent Text Processing*, pages 19–25.

Csomai, A. and Mihalcea, R. (2007). Investigations in Unsupervised Back-of-the-book Indexing. In *Proceedings of the Florida Artificial Intelligence Research Society*, pages 211–216.

Csomai, A. and Mihalcea, R. (2008). Linguistically Motivated Features for Enhanced Back-of-the-Book Indexing. In *Proceedings of the Association for Computational Linguistics (ACL 2008)*, pages 932–940.

Cucerzan, S. (2007). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *EMNLP-CoNLL*, volume 2007, pages 708–716.

Damianos, L., Cuomo, D., Griffith, J., Hirst, D., and Smallwood, J. (2007). Exploring the Adoption, Utility, and Social Influences of Social Bookmarking in a Corporate Environment. In *40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 86–86.

Daumé, H. and Marcu, D. (2005). Learning as Search Optimization: Approximate Large Margin Methods for Structured Prediction. *Proceedings of the 22nd International Conference on Machine Learning*, pages 169–176.

Daxenberger, J. and Gurevych, I. (2013). Automatically Classifying Edit Categories in Wikipedia Revisions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 578–589.

Désilets, A., Paquet, S., and Vinson, N. G. (2005). Are Wikis Usable? In *Proceedings of the 2005 International Symposium on Wikis - WikiSym '05*, pages 3–15.

Dinkelacker, J. and Garg, P. (2001). Corporate Source: Applying Open Source Concepts to a Corporate Environment. In *Proceedings of the First Workshop on Open Source Software Engineering*.

Dörner, C., Pipek, V., and Won, M. (2007). Supporting Expertise Awareness: Finding Out What Others Know. In *Proceedings of the 2007 Symposium on Computer Human Interaction for the Management of Information Technology*, pages 9–18.

Dresner, E. and Herring, S. C. (2010). Functions of the Nonverbal in CMC: Emoticons and Illocutionary Force. *Communication Theory*, 20(3):249–268.

Drouin, M. and Davis, C. (2009). R u Txting? Is the Use of Text Speak Hurting your Literacy? *Journal of Literacy Research*, 41(1):46–67.

Durme, B. V. and Osborne, M. (2014). Exponential Reservoir Sampling for Streaming Language Models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 687–692.

Eckart de Castilho, R. (2014). *Natural Language Processing: Integration of Automatic and Manual Analysis*. Dissertation, Technische Universität Darmstadt.

Eckart de Castilho, R. and Gurevych, I. (2009). DKPro-UGD: A Flexible Data-Cleansing Approach to Processing User-Generated Discourse. In *Online Proceedings of the First French-speaking Meeting around the Framework Apache UIMA*.

Eckart de Castilho, R. and Gurevych, I. (2011). A Lightweight Framework for Reproducible Parameter Sweeping in Information Retrieval. In *Proceedings of the 2011 Workshop on Data Infrastructures for Supporting Information Retrieval Evaluation*, pages 7–10.

Eckart de Castilho, R. and Gurevych, I. (2014). A Broad-coverage Collection of Portable NLP Components for Building Shareable Analysis Pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT) at COLING 2014*, pages 1–11.

Eisenstein, J. (2013). What to Do About Bad Language on the Internet. In *HLT-NAACL*,

pages 359–369.

Erbs, N., Agirre, E., Soroa, A., Barrena, A., Gurevych, I., and Zesch, T. (2012). UKP-UBC Entity Linking at TAC-KBP. In *Proceedings of Text Analysis Conference (TAC 2012)*, Gaithersburg, Maryland USA.

Erbs, N., Bär, D., Gurevych, I., and Zesch, T. (2011a). First Aid for Information Chaos in Wikis: Collaborative Information Management Enhanced Through Language Technology. In *Information und Wissen: Global, Sozial und Frei? : Proceedings des 12. Internationalen Symposiums für Informationswissenschaften*, pages 501–502.

Erbs, N., Gurevych, I., and Rittberger, M. (2013a). Bringing Order to Digital Libraries: From Keyphrase Extraction to Index Term Assignment. *D-Lib Magazine*, 19(9/10):1–16.

Erbs, N., Gurevych, I., and Zesch, T. (2013b). Hierarchy Identification for Automatically Generating Table-of-Contents. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 252–260.

Erbs, N., Gurevych, I., and Zesch, T. (2014a). Sense and Similarity : A Study of Sense-level Similarity Measures. In *Proceedings of the 3rd Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 30–39.

Erbs, N., Santos, P. B., Gurevych, I., and Zesch, T. (2014b). DKPro Keyphrases : Flexible and Reusable Keyphrase Extraction Experiments. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. System Demonstrations*, pages 31–36.

Erbs, N., Zesch, T., and Gurevych, I. (2011b). Link Discovery: A Comprehensive Analysis. In *Proceedings of the 5th IEEE International Conference on Semantic Computing (IEEE-ICSC)*, pages 83–86.

Erkan, G. and Radev, D. (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

Feng, J., Haffner, P., and Gilbert, M. (2005). A Learning Approach to Discovering Web Page Semantic Structures. In *Proceedings of the Eight International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 1055–1059.

Ferret, O. (2007). Finding Document Topics for Improving Topic Segmentation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 480–487, Prague, Czech Republic.

Ferrucci, D., Brown, E., Chu-carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., and Prager, J. (2010). Building Watson : An Overview of the DeepQA Project. *AI Magazine*, 31(3):59–79.

Ferrucci, D. and Lally, A. (2004). UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering*, 10(3-4):327–348.

Ferschke, O., Daxenberger, J., and Gurevych, I. (2013). A Survey of NLP Methods and Resources for Analyzing the Collaborative Writing Process in Wikipedia. In *The People's Web Meets NLP: Collaboratively Constructed Language Resources*.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing Search in Context: The Concept Revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Fischer-Starcke, B. (2010). *Corpus Linguistics in Literary Analysis: Jane Austen and her Contemporaries*. Continuum.

Fisher, R. (1915). Frequency Distribution of the Values of the Correlation Coefficient in Samples from an Indefinitely Large Population. *Biometrika*, pages 507–521.

Fisher, R. (1921). On the "Probable Error" of a Coefficient of Correlation Deduced from a Small Sample. *Metron*, 1:3–32.

Flanagin, A. and Metzger, M. (2001). Internet Use in the Contemporary Media Environment. *Human Communication Research*, 27(1):153–181.

Flekova, L., Ferschke, O., and Gurevych, I. (2014). What Makes a Good Biography? Multidimensional Quality Analysis Based on Wikipedia Article Feedback Data. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 855–866.

Fowler, R. L. and Barker, A. S. (1974). Effectiveness of Highlighting for Retention of Text Material. *Journal of Applied Psychology*, 59(3):358.

Frank, E., Paynter, G. W., Witten, I., Gutwin, C., and Nevill-Manning, C. G. (1999). Domain-specific Keyphrase Extraction. In *Proceedings of the 16th International Joint Conference on Aritificial Intelligence (IJCAI)*, pages 668–673.

Fürnkranz, J. (1999). Exploiting Structural Information for Text Classification on the WWW. *Advances in Intelligent Data Analysis*, pages 487–497.

Gabrilovich, E. and Markovitch, S. (2007). Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 1606–1611.

Getoor, L. and Diehl, C. P. (2005). Link Mining: A Survey. *SigKDD Explorations Special Issue on Link Mining*, 7(2):3–12.

Geva, S. (2007). GPX: Ad-Hoc Queries and Automated Link Discovery in the Wikipedia. In *Preproceedings of the INEX Workshop*, pages 404–416.

Giles, J. (2005). Internet Encyclopaedias Go Head to Head. *Nature*, 438(7070):900–901.

Goldstein, J., Mittal, V., Carbonell, J., and Kantrowitz, M. (2000). Multi-Document Summarization By Sentence Extraction. In *Proceedings of the NAACL-ANLP 2000 Workshop: Automatic Summarization*, pages 40–48.

Gordon, C. A. (1995). *Concept Mapping as a Pre-search Activity in the Research Process*. Boston University.

Greene, W. H. (2003). *Econometric Analysis*. Pearson Education India.

Grefenstette, G. (1992). Sextant: Exploring Unexplored Contexts for Semantic Extraction from Syntactic Analysis. In *Proceedings of the 30th Annual Meeting of the*

Association for Computational Linguistics, pages 324–326, Newark, Delaware, USA.

Guillory, H. (1998). The Effects of Keyword Captions to Authentic French Video on Learner Comprehension. *Calico Journal*, 15:89–108.

Gurevych, I., Eckle-Kohler, J., Hartmann, S., Matuschek, M., Meyer, C. M., and Wirth, C. (2012). UBY–A Large-Scale Unified Lexical-Semantic Resource Based on LMF. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 580–590.

Gurevych, I., Mühlhäuser, M., Müller, C., Steimle, J., Weimer, M., and Zesch, T. (2007a). Darmstadt Knowledge Processing Repository Based on UIMA. In *Proceedings of the First Workshop on Unstructured Information Management Architecture at Biannual Conference of the Society for Computational Linguistics and Language Technology*.

Gurevych, I., Müller, C., and Zesch, T. (2007b). What to be? - Electronic Career Guidance Based on Semantic Relatedness. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 1032–1039.

Gurevych, I. and Wolf, E. (2010). Expert-built and Collaboratively Constructed Lexical Semantic Resources. *Language and Linguistics Compass*, 4(11):1074–1090.

Gutwin, C. (1999). Improving Browsing in Digital Libraries with Keyphrase Indexes. *Decision Support Systems*, 27(1-2):81–104.

Hachey, B., Radford, W., and Curran, J. R. (2011). Graph-based Named Entity Linking with Wikipedia. In *Proceedings of the 12th International Conference on Web Information System Engineering*, WISE'11, pages 213–226.

Hachey, B., Radford, W., Nothman, J., Honnibal, M., and Curran, J. R. (2012). Evaluating Entity Linking with Wikipedia. *Artificial Intelligence*, 194:130–150.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA Data Mining Software: an Update. *SIGKDD Explorations*, 11(1):10–18.

Haller, H. and Abecker, A. (2010). iMapping: a Zooming User Interface Approach for Personal and Semantic Knowledge Management. In *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, pages 119–128.

Hamp, B. and Feldweg, H. (1997). GermaNet - A Lexical-Semantic Net for German. In *Proceedings of ACL Workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15.

Han, X. and Sun, L. (2011). A Generative Entity-mention Model for Linking Entities with Knowledge Base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 945–954.

Han, X., Sun, L., and Zhao, J. (2011). Collective Entity Linking in Web Text: A Graph-based Method. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 765–774.

Han, X. and Zhao, J. (2009). NLPR_KBP in TAC 2009 KBP Track: A Two-stage Method to Entity Linking. In *Proceedings of Text Analysis Conference (TAC 2009)*.

Harnisch, S. and Knaf, S. (2014). Exploring Tariff-choice Preferences in B2B Enterprise Software Acquisition Settings. In *Proceedings of the European Conference on*

Information Systems (ECIS), pages 1–16.

Hartley, J., Bartlett, S., and Branthwaite, A. (1980). Underlining Can Make a Difference: Sometimes. *The Journal of Educational Research*, 13:218–224.

Hasan, K. and Ng, V. (2010). Conundrums in Unsupervised Keyphrase Extraction: Making Sense of the State-of-the-art. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 365–373.

Hasan, K. S. and Ng, V. (2014). Automatic Keyphrase Extraction: A Survey of the State of the Art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1262–1273.

Hassan, S. and Mihalcea, R. (2011). Semantic Relatedness Using Salient Semantic Analysis. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence, (AAAI 2011)*, pages 884–889, San Francisco, CA, USA.

Hearst, M. A. (1993). TextTiling: A Quantitative Approach to Discourse Segmentation.

Hoffart, J., Bär, D., Zesch, T., and Gurevych, I. (2009). Discovering Links Using Semantic Relatedness. In *Preproceedings of the INEX Workshop*, pages 314–325, Brisbane, Australia.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust Disambiguation of Named Entities in Text. In *Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

Hofland, K. and Johansson, S. (1982). *Word Frequencies in British and American English*. Norwegian Computing Centre for the Humanities.

Hollink, V., Kamps, J., Monz, C., and de Rijke, M. (2004). Monolingual Document Retrieval for European Languages. *Information Retrieval*, 7(1/2):33–52.

Hölscher, C. and Strube, G. (2000). Web Search Behavior of Internet Experts and Newbies. *Computer Networks*, 33:337–346.

Hooper, J. B. (1979). Substantive Principles in Natural Generative Phonology. *Current Approaches to Phonological Theory*, pages 106–125.

Huang, D. W. C., Geva, S., and Trotman, A. (2008). Overview of the INEX 2008 Link the Wiki Track. In *Proceedings of the 7th International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 314–325.

Huang, D. W. C., Geva, S., and Trotman, A. (2009a). Overview of the INEX 2009 Link the Wiki Track. In *Proceedings of the 8th International Workshop of the Initiative for the Evaluation of XML Retrieval*, Lecture Notes in Computer Science, pages 312–323.

Huang, D. W. C., Trotman, A., and Geva, S. (2009b). The Importance of Manual Assessment in Link Discovery. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '09*, pages 698–699.

Huang, D. W. C., Xu, Y., Trotman, A., and Geva, S. (2007). Overview of INEX 2007 Link the Wiki Track. In *Proceedings of the 6th International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 373–387.

Huang, H.-S., Chiou, C.-C., Chiang, H.-K., Lai, S.-H., Huang, C.-Y., and Chou, Y.-Y. (2012). Effects of Multidimensional Concept Maps on Fourth Graders' Learning in Web-based Computer Course. *Computers & Education*, 58(3):863–873.

Hughes, T. P. (2004). *Human-built World: How to Think about Technology and Culture*. University of Chicago Press.

Hulth, A. (2003). Improved Automatic Keyword Extraction Given More Linguistic Knowledge. In *Proceedings of the 2003 Conference on Empirical Methods for Natural Language Processing*, pages 216–223.

Hulth, A. (2004). Enhancing Linguistically Oriented Automatic Keyword Extraction. In *Proceedings of HLT-NAACL: Short Papers*, pages 17–20.

Hurtienne, J. and Wandke, H. (1997). Wie Effektiv und Effizient Navigieren Benutzer im World Wide Web? Eine Empirische Studie. In *CAW-97: Beiträge zum Workshop Cognition & Web. Freiburg: IIG Berichte*, volume 1, pages 93–104.

Itakura, K. Y. and Clarke, C. L. A. (2007). University of Waterloo at INEX2007: Adhoc and Link-the-Wiki Tracks. In *Proceedings of 6th International Workshop of the Initiative for the Evaluation of XML Retrieval*, volume 4862, pages 417–425.

Jäschke, R. and Marinho, L. (2007). Tag Recommendations in Folksonomies. In *Proceedings of the International Workshop at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 506–514.

Jelinek, F. and Mercer, R. L. (1980). Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Pattern Recognition in Practice*, pages 381–397.

Ji, H., Grishman, R., and Dang, H. T. (2011). Overview of the TAC 2011 Knowledge Base Population Track. In *Proceedings of Text Analysis Conference (TAC 2011)*.

Ji, H., Grishman, R., Dang, H. T., Griffitt, K., and Ellis, J. (2010). Overview of the TAC 2010 Knowledge Base Population Track. In *Proceedings of Text Analysis Conference (TAC 2010)*.

Jiang, J. J. and Conrath, D. W. (1997). Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of 10th International Conference Research on Computational Linguistics*, pages 1–15.

Jin, R. and Hauptmann, A. (2001). Automatic Title Generation for Spoken Broadcast News. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–3.

Jurgens, D. (2014). An Analysis of Ambiguity in Word Sense Annotations. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 3006–3012.

Kaplan, A. M. and Haenlein, M. (2010). Users of the World, Unite! The Challenges and Opportunities of Social Media. *Business Horizons*, 53(1):59–68.

Kettunen, K., Kunttu, T., and Järvelin, K. (2005). To Stem or Lemmatize a Highly Inflectional Language in a Probabilistic IR Environment? *Journal of Documentation*, 61(4):476–496.

Kiewra, K. A., Benton, S. L., Kim, S.-I., Risch, N., and Christensen, M. (1995). Effects

of Note-taking Format and Study Technique on Recall and Relational Performance. *Contemporary Educational Psychology*, 20(2):172–187.

Kiewra, K. A., DuBois, N. F., Christian, D., and McShane, A. (1988). Providing Study Notes: Comparison of Three Types of Notes for Review. *Journal of Educational Psychology*, 80(4):595.

Kim, S. N., Medelyan, O., Kan, M. Y., and Baldwin, T. (2010). SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26.

Kim, S. N., Medelyan, O., Kan, M.-Y., and Baldwin, T. (2013). Automatic Keyphrase Extraction from Scientific Articles. *Language Resources and Evaluation*, 47:723–742.

Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. (2008). A Large-scale Classification of English Verbs. *Language Resources and Evaluation*, 42(1):21–40.

Kittur, A., Suh, B., Pendleton, B., and Chi, E. (2007). He Says, She Says: Conflict and Coordination in Wikipedia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 453–462.

Kleinberg, J. M. (1999). Authoritative Sources in a Hyperlinked Environment. In *Journal of the ACM*, volume 46, pages 604–632.

Koehn, P. and Knight, K. (2003). Empirical Methods for Compound Splitting. *Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics*, pages 187–193.

Kommers, P. and Lanzing, J. (1997). Students' Concept Mapping for Hypermedia Design: Navigation through World Wide Web (WWW) Space and Self-Assessment. *Journal of Interactive Learning Research*, 8:421–455.

Kopak, R. W. (2000). *A Taxonomy of Link Types for Use in Hypertext*. PhD thesis, Faculty of Information Studies, University of Toronto.

Kozima, H. (1993). Text Segmentation based on Similarity between Words. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, pages 286–288, Morristown, NJ, USA.

Krippendorff, K. (2012). *Content Analysis: An Introduction to its Methodology*. Sage.

Kuhlthau, C. and Tama, S. (2001). Information Search Process of Lawyers: a Call for 'Just for Me' Information Services. *Journal of Documentation*, 57(1):25–43.

Kuhlthau, C. C. (1993). *Seeking Meaning: A Process Approach to Library and Information Services*. Ablex Norwood, NJ.

Kulkarni, S., Singh, A., Ramakrishnan, G., and Chakrabarti, S. (2009). Collective Annotation of Wikipedia Entities in Web Text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '09*, pages 457–465, New York, New York, USA. ACM Press.

Kupietz, M. and Belica, C. (2010). The German Reference Corpus DeReKo: A Primordial Sample for Linguistic Research. In *Proceedings of the 7th Conference on International Language Resources and Evaluation*, pages 1848–1854.

Kwok, C., Etzioni, O., and Weld, D. S. (2001). Scaling Question Answering to the Web.

ACM Transactions on Information Systems, 19(3):242–262.

Ladd, P. R. (2009). The Wikipedia Revolution: How A Bunch of Nobodies Created The World's Greatest Encyclopedia. *International Journal of Knowledge Content Development & Technology*, 1(2):53–54.

Ladendorf, O. (1906). *Historisches Schlagwörterbuch*. Karl J Trübner.

Langer, S. (1998). Zur Morphologie und Semantik von Nominalkomposita. In *Tagungsband der 4. Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS)*, pages 83–97.

Lanzing, J. W. A. (1998). Concept Mapping: Tools for Echoing the Minds Eye. *Journal of Visual Literacy*, 18(1).

Larson, M. and Willett, D. (2000). Compound Splitting and Lexical Unit Recombination for Improved Performance of a Speech Recognition System for German Parliamentary Speeches. *INTERSPEECH*, pages 945–948.

Lehmann, J., Monahan, S., Nezda, L., Jung, A., and Shi, Y. (2010). LCC Approaches to Knowledge Base Population at TAC 2010. In *Proceedings of Text Analysis Conference (TAC 2010)*.

Lepp, F. (1908). *Schlagwörter des Reformationszeitalters*. M. Heinsius Nachfolger.

Lesk, M. (1986). Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24–26.

Leuf, B. and Cunningham, W. (2001). *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley New York.

Levin, B. (1993). English Verb Classe and Alternations: A Preliminary Investigation.

Li, D., Browne, G., and Wetherbe, J. (2006). Why do Internet Users Stick with a Specific Web Site? A Relationship Perspective. *International Journal of Electronic Commerce*, 10(4):105–141.

Lin, D. (1998). An Information-theoretic Definition of Similarity. In *Proceedings of the International Conference on Machine Learning*, volume 98, pages 296–304.

Lin, D. and Pantel, P. (2001). Discovery of Inference Rules for Question-answering. *Natural Language Engineering*, 7(4):343–360.

Lipczak, M. (2008). Tag Recommendation for Folksonomies Oriented towards Individual Users. *Proceedings of the International Workshop at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 84–95.

Lopez, C., Prince, V., and Roche, M. (2011). Automatic Titling of Articles Using Position and Statistical Information. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 727–732.

Lopez, P. and Romary, L. (2010). HUMB: Automatic Key Term Extraction from Scientific Articles in GROBID. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 248–251.

Lovins, J. (1968). Development of a Stemming Algorithm. *Mechanical Translation and*

Computational Linguistics, 11(6):22–31.

Lyons, J. (1977). Semantics, Volume I. *Cambridge UP, Cambridge*.

Madjarov, G., Kocev, D., Gjorgjevikj, D., and Džeroski, S. (2012). An Extensive Experimental Comparison of Methods for Multi-label Learning. *Pattern Recognition*, 45:3084–3104.

Majchrzak, A., Wagner, C., and Yates, D. (2006). Corporate Wiki Users: Results of a Survey. In *Proceedings of the International Symposium on Wikis (WikiSym)*, pages 99–104.

Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *An Introduction to Information Retrieval*. Cambridge University Press Cambridge.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Marek, T. (2006). Analysis of German Compounds using Weighted Finite State Transducers. *Bachelor thesis, University of Tübingen*.

Matthews, P. H. (1972). *Inflectional Morphology: A Theoretical Study based on Aspects of Latin Verb Conjugation*, volume 6. Cambridge University Press.

McCandless, M., Hatcher, E., and Gospodnetic, O. (2010). *Lucene in Action: Covers Apache Lucene 3.0*. Manning Publications Co.

McDaniel, M. A. and Pressley, M. (1989). Keyword and Context Instruction of New Vocabulary Meanings: Effects on Text comprehension and Memory. *Journal of Educational Psychology*, 81(2):204–213.

McNamee, P. and Dang, H. T. (2009). Overview of the TAC 2009 Knowledge Base Population Track. In *Proceedings of Text Analysis Conference (TAC 2009)*.

McNamee, P., Dang, H. T., Simpson, H., Schone, P., and Strassel, S. M. (2010). An Evaluation of Technologies for Knowledge Base Population. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 369–372.

McNemar, Q. (1947). Note on the Sampling Error of the Difference between Correlated Proportions or Percentages. *Psychometrika*, 12(2):153–157.

Medelyan, O., Frank, E., and Witten, I. H. (2009). Human-competitive Tagging using Automatic Keyphrase Extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1318–1327.

Medelyan, O. and Witten, I. H. (2006). Thesaurus based Automatic Keyphrase Indexing. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital libraries*, pages 296–297.

Medelyan, O., Witten, I. H., and Milne, D. (2008). Topic Indexing with Wikipedia. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy*, pages 19–24, Chicago, USA.

Mendes, P. N., Jakob, M., and Bizer, C. (2011). DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8.

Meyer, C., Mieskes, M., Stab, C., and Gurevych, I. (2014). DKPro Agreement: An Open-Source Java Library for Measuring Inter-Rater Agreement. In *Proceedings of the the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1501–1510.

Meyer, C. M. and Gurevych, I. (2012a). To Exhibit is not to Loiter: A Multilingual, Sense-disambiguated Wiktionary for Measuring Verb Similarity. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1763–1780, Mumbai, India.

Meyer, C. M. and Gurevych, I. (2012b). Wiktionary: A New Rival for Expert-built Lexicons? Exploring the Possibilities of Collaborative Lexicography. In *Electronic Lexicography*, chapter 13, pages 259–291. Oxford, UK.

Mihalcea, R. and Csomai, A. (2007). Wikify!: Linking Documents to Encyclopedic Knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM*, pages 233–242, Lisbon, Portugal.

Mihalcea, R. and Tarau, P. (2004a). TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

Mihalcea, R. and Tarau, P. (2004b). TextRank: Bringing Order into Texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

Mikros, G. and Argiri, E. K. (2007). Investigating Topic Influence in Authorship Attribution. In *Proceedings of the SIGIR 2007 International Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection, PAN 2007*.

Miller, G. A. and Charles, W. G. (1991). Contextual Correlates of Semantic Similarity. *Language and Cognitive Processes*, 6(1):1–28.

Miller, T. and Gurevych, I. (2014). WordNet–Wikipedia–Wiktionary: Construction of a Three-way Alignment. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 2094–2100.

Milne, D. (2007). Computing Semantic Relatedness using Wikipedia Link Structure. In *Proceedings of the New Zealand Computer Science Research Student Conference*.

Milne, D. and Witten, I. (2008a). An Effective, Low-cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, pages 25–30.

Milne, D. and Witten, I. H. (2008b). Learning to Link with Wikipedia. In *CIKM '08: Proceeding of the 17th ACM Conference on Information and Knowledge Management*, pages 509–518, New York, USA.

Monahan, S., Lehmann, J., and Nyberg, T. (2011). Cross-lingual Cross-document Coreference with Entity Linking. In *Proceedings of Text Analysis Conference (TAC 2011)*.

Morris, J. and Hirst, G. (1991). Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. *Computational Linguistics*, 17:21–48.

Munson, S. (2008). Motivating and Enabling Organizational Memory with a Workgroup Wiki. In *Proceedings of the 4th International Symposium on Wikis*, pages 18–23.

Nakayama, K., Hara, T., and Nishio, S. (2007). Wikipedia Mining for an Association Web Thesaurus Construction. In *Web Information Systems Engineering – WISE 2007*, Lecture Notes in Computer Science, pages 322–334.

Navigli, R. (2009). Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, 41(2):1–69.

Navigli, R. and Ponzetto, S. P. (2012). An Overview of BabelNet and its API for Multilingual Language Processing. In *The People's Web Meets NLP: Collaboratively Constructed Language Resources*, pages 177–197.

Nguyen, T. D. and Kan, M.-Y. (2007). Keyphrase Extraction in Scientific Publications. In *Proceedings of International Conference on Asian Digital Libraries*, volume 4822 of *Lecture Notes in Computer Science*, pages 317–326.

Nguyen, V. C., Nguyen, L. M., and Shimazu, A. (2009). A Semi-supervised Approach for Generating a Table-of-contents. In *Proceedings of the International Conference RANLP-2009*, pages 312–317.

Nie, N. and Erbring, L. (2000). Internet and Society. *Stanford Institute for the Quantitative Study of Society*.

Nielsen, J. (1997). How Users Read on the Web. *Jakob Nielsen's Alertbox*, 20:4–7.

Nist, S. L. and Hogrebe, M. C. (1987). The Role of Underlining and Annotating in Remembering Textual Information. *Literacy Research and Instruction*, 27(1):12–25.

Norman, D. (2002). Emotion Design: Attractive Things Work Better. *Interactions*, 9(4):36–42.

Nov, O. (2007). What Motivates Wikipedians? *Communications of the ACM*, 50(11):60–64.

Novak, J. D. and Cañas, A. J. (2008). The Theory Underlying Concept Maps and How to Construct and Use Them. *Florida Institute for Human and Machine Cognition Pensacola*, 284:1–58.

Nylander, S., Lundquist, T., and Brännström, A. (2009). At Home and with Computer Access – Why and Where People Use Cell Phones to Access the Internet. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1639–1642.

Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Okumura, M. and Honda, T. (1994). Word Sense Disambiguation and Text Segmentation Based On Lexical Cohesion. In *Proceedings of the 15th Conference on Computational linguistics*, pages 755–761.

Ordelman, R. J. F. (2003). *Dutch Speech Recognition in Multimedia Information Retrieval*. PhD thesis, University of Twente, Enschede, Enschede.

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., and Ioma, C. (2006). Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of the Open Source Information Retrieval Workshop*, pages 18–25.

Over, P. and Yen, J. (2004). Introduction to DUC-2001: An Intrinsic Evaluation of Generic News Text Summarization Systems. In *Proceedings of DUC 2004 Document Understanding Workshop*, Boston, USA.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab.

Parunak, H. V. D. (1991). Ordering the Information Graph. *Hypertext/hypermedia handbook*, pages 299–325.

Pedersen, T., Purandare, A., and Kulkarni, A. (2005). Name Discrimination by Clustering Similar Contexts. In *Computational Linguistics and Intelligent Text Processing*, pages 226–237.

Peirce, C. S. (1974). *Collected Papers of Charles Sanders Peirce*. Harvard University Press.

Peirce, C. S. S. (1906). Prolegomena to an Apology for Pragmaticism. *The Monist*, 16(4):492–546.

Pembe, F. and Güngör, T. (2010). A Tree Learning Approach to Web Document Sectional Hierarchy Extraction. In *Proceedings of 2nd International Conference on Agents and Artificial Intelligence*, pages 447–450.

Pfaffenberger, B. (1992). Social Anthropology of Technology. *Annual Review of Anthropology*, 21(1):491–516.

Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program: Electronic Library and Information Systems*, 14(3):130–137.

Press, W. H. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press.

Purver, M., Griffiths, T. L., Körding, K. P., and Tenenbaum, J. B. (2006). Unsupervised Topic Modelling for Multi-party Spoken Discourse. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 17–24, Morristown, NJ, USA.

Quinlan, J. R. (1992). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

Rao, D., McNamee, P., and Dredze, M. (2013). Entity linking: Finding Extracted Entities in a Knowledge Base. In *Multi-source, Multilingual Information Extraction and Summarization*, pages 93–115.

Ravid, G., Kalman, Y., and Rafaeli, S. (2008). Wikibooks in Higher Education: Empowerment through Online Distributed Collaboration. *Computers in Human Behavior*, 24(5):1913–1928.

Rayson, P. and Garside, R. (2000). Comparing Corpora Using Frequency Profiling. In *Proceedings of the Workshop on Comparing Corpora*, pages 1–6.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier Chains for

Multi-label Classification. *Machine Learning*, 85(3):333–359.

Resnik, P. (1999). Semantic Similarity in a Taxonomy: An Information-based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, 11:95–130.

Richter, A. and Riemer, K. (2009). Corporate Social Networking Sites – Modes of Use and Appropriation through Co-Evolution. In *Proceedings of the 20th Australasian Conference on Information Systems*.

Robinson, M. A. (2010). An Empirical Analysis of Engineers ' Information Behaviors. *Journal of the American Society for Information Science and Technology*, 61(4):640–658.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual Correlates of Synonymy. *Communications of the ACM*, 8(10):627–633.

Rudin, C. and Wagstaff, K. L. (2013). Machine Learning for Science and Society. *Machine Learning*, 95(1):1–9.

Ruppenhofer, J., Ellsworth, M., Petruck, M. R. L., Johnson, C. R., and Scheffczyk, J. (2010). *FrameNet II: Extended Theory and Practice*. International Computer Science Institute.

Salton, G., Allan, J., and Buckley, C. (1993). Approaches to Passage Retrieval in Full Text Information Systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58.

Salton, G. and Buckley, C. A. (1988). Term-weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5):513–523.

Sauper, C. and Barzilay, R. (2009). Automatically Generating Wikipedia Articles: A Structure-aware Approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216.

Schiller, A. (2006). German Compound Analysis with wfsc. In *Finite-State Methods and Natural Language Processing*, pages 239–246.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.

Schneckenberg, D. (2009). Web 2.0 and the Empowerment of the Knowledge Worker. *Journal of Knowledge Management*, 13(6):509–520.

Schwartz, H. A. and Gomez, F. (2011). Evaluating Semantic Metrics on Tasks of Concept Similarity. In *Proceedings of the International Conference of the Florida Artificial Intelligence Research Society*.

Schwarz, C. K. (2010). Effektivere Informationssuche im World Wide Web mit Hilfe von Vorschau-Fenstern (Overview Snippets): Eine Experimentelle Überprüfung. B.sc. thesis, TU Darmstadt, Institute of Psychology.

Schwarz, C. K., Keith, N., Gurevych, I., Erbs, N., and Zesch, T. (2010). Effektivere Informationssuche im World Wide Web. KCreativity, Learning Strategies and

Efficiency in E-learning 2010 Poster Presentation.

Scott, M. (1996). WordSmith Tools Manual.

Scott, M. (1997). PC Analysis of Key Words—and Key Key Words. *System*, 25(2):233–245.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47.

Seco, N., Veale, T., and Hayes, J. (2004). An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In *Proceedings of European Conference for Artificial Intelligence*, pages 1089–1093.

Shapiro, C. and Varian, H. R. (2013). *Information Rules: A Strategic Guide to the Network Economy*. Harvard Business Press.

Shen, L. and Joshi, A. K. (2005). Ranking and Reranking with Perceptron. *Machine Learning*, 60(1-3):73–96.

Shen, W., Wang, J., Luo, P., and Wang, M. (2012). LINDEN: Linking Named Entities with Knowledge Base via Semantic Knowledge. In *Proceedings of the 21st International Conference on World Wide Web*, pages 449–458.

Sinclair, J. (1991). *Corpus, Concordance, Collocation*. Oxford University Press Oxford, 1st edition.

Singer, G., Pruulmann-Vengerfeldt, P., Norbisrath, U., and Lewandowski, D. (2012). The Relationship Between Internet User Type and User Performance when Carrying out Simple vs. Complex Search Tasks. *First Monday*, 17(6).

Slonim, N., Friedman, N., and Tishby, N. (2002). Unsupervised Document Classification Using Sequential Information Maximization. In *Proceedings of the 25th International Conference on Research and Development in Information Retrieval*, pages 129–136.

Song, M., Song, I. Y., Allen, R. B., and Obradovic, Z. (2006). Keyphrase Extraction-based Query Expansion in Digital Libraries. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 202–209.

Sparck Jones, K. and van Rijsbergen, C. (1975). Report on the Need for and Provision of an 'Ideal' Information Retrieval Test Collection. Technical report, Computer Laboratory, University of Cambridge.

Spitkovsky, V. and Chang, A. (2012). A Cross-lingual Dictionary for English Wikipedia Concepts. In *Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3168–3175.

Spyromitros, E., Tsoumakas, G., and Vlahavas, I. (2008). An Empirical Study of Lazy Multilabel Classification Algorithms. In *Artificial Intelligence: Theories, Models and Applications*, pages 401–406.

Stanovich, K. E. (2000). *Progress in Understanding Reading: Scientific Foundations and New Frontiers*. Guilford Press.

Stein, K. and Hess, C. (2007). Does It Matter Who Contributes: A Study on Featured Articles in the German Wikipedia. In *HT '07: Proceedings of the Eighteenth Conference on Hypertext and Hypermedia*, pages 171–174, New York, NY, USA.

Stille, W., Erbs, N., Zesch, T., Gurevych, I., and Weihe, K. (2011). Aufbereitung und Strukturierung von Information mittels automatischer Sprachverarbeitung. In *Proceedings of KnowTech*, pages 199–208.

Strube, M. and Ponzetto, S. (2006). WikiRelate! Computing Semantic Relatedness using Wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 1419–1424.

Sunercan, O. and Birturk, A. (2010). Wikipedia Missing Link Discovery: A Comparative Study. In *Proceedings of AAAI Spring Symposium on Linked Data Meets Artificial Intelligence (Linked AI 2010)*, pages 126–131.

Sweet, A. P. and Snow, C. E. (2003). *Rethinking Reading Comprehension. Solving Problems in the Teaching of Literacy*. ERIC.

Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science*, 12(2):257–285.

Taşkin, M., Pepe, H., Taşkin, C., Gevat, C., and Taşkin, H. (2011). The Effect of Concept Maps in Teaching Sportive Technique. *Procedia - Social and Behavioral Sciences*, 11:141–144.

Tergan, S. (2004). Concept Maps for Managing Individual Knowledge. In *Proceedings of the First Joint Meeting of the EARLI SIGS*, pages 229–238.

Tewksbury, D. (2003). What do Americans Really Want to Know? Tracking the Behavior of News Readers on the Internet. *Journal of Communication*, 53(12):694–710.

Tognini-Bonelli, E. (2001). *Corpus Linguistics at Work*, volume 6. John Benjamins Publishing.

Toman, M., Tesar, R., and Jezek, K. (2006). Influence of Word Normalization on Text Classification. In *Proceedings of InSciT*, pages 354–358.

Tomokiyo, T. and Hurst, M. (2003). A Language Model Approach to Keyphrase Extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 33–40.

Toolan, M. (2004). Values are Descriptions; or, from Literature to Linguistics and back again by Way of Keywords: The Linguistics: Literature Lnterface. *Belgian Journal of English Language and Literatures*, pages 11–30.

Toutanova, K. and Klein, D. (2003). Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 173–180.

Toutanova, K. and Manning, C. (2000). Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and very Large Corpora*, pages 63–70.

Treem, J. W. and Leonardi, P. M. (2012). Social Media Use in Organizations. *Communication Yearbook*, 36:143–189.

Trigg, R. H. (1983). *A Network-based Approach to Text Handling for the Online*

Scientific Community. PhD thesis, University of Maryland, College Park.

Trotman, A., Alexander, D., and Geva, S. (2010). Overview of the INEX 2010 Link the Wiki Track. In *Proceedings of the 9th International Workshop of the Inititative for the Evaluation of XML Retrieval*, pages 241–249.

Tsoumakas, G. and Katakis, I. (2007). Multi-label Classification: an Overview. *International Journal of Data Warehousing and Mining*, 3(9):1–13.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010). Mining Multi-label Data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2011). Random k-labelsets: An Ensemble Method for Multilabel Classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):1079–1089.

Tucker, S. and Whittaker, S. (2009). Have A Say Over What You See: Evaluating Interactive Compression Techniques. In *Proceedings of the 2009 International Conference on Intelligent User Interfaces*, pages 37–46.

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, pages 433–460.

Turney, P. D. (2000). Learning Algorithms for Keyphrase Extraction. *Information Retrieval*, 2(4):303–336.

Vakkari, P. (2000). Relevance and Contributing Information Types of Searched Documents in Task Performance. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2–9.

Van Rijsbergen, C. J., Robertson, S. E., and Porter, M. F. (1980). *New Models in Probabilistic Information Retrieval*. Computer Laboratory, University of Cambridge.

Varma, V., Bharat, V., Kovelamudi, S., Bysani, P., N, K. K., Reddy, K., Kumar, K., and Maganti, N. (2009). IIIT Hyderabad at TAC 2009. Technical report.

Véronis, J. (1998). A Study of Polysemy Judgements and Inter-annotator Agreement. In *Programme and Advanced Papers of the SensEval Workshop*, pages 2–4.

Vickery, G. and Wunsch-Vincent, S. (2007). *Participative Web and User-created Content: Web 2.0 Wikis and Social Networking*. Organization for Economic Cooperation and Development (OECD).

von Restorff, H. (1933). Über die Wirkung von Bereichsbildungen im Spurenfeld. *Psychologische Forschung*, 18(1):299–342.

Wagstaff, K. (2012). Machine Learning that Matters. In *Proceedings of the Twenty-Ninth International Conference on Machine Learning (ICML)*, pages 529–536.

Walther, J. B. and D'Addario, K. P. (2001). The Impacts of Emoticons on Message Interpretation in Computer-mediated Communication. *Social Science Computer Review*, 19(3):324–347.

Wan, X. and Xiao, J. (2008a). CollabRank: Towards a Collaborative Approach to Single-document Keyphrase Extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 969–976.

Wan, X. and Xiao, J. (2008b). Single Document Keyphrase Extraction Using

Neighborhood Knowledge. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 855–860, Chicago, USA.

Wang, L. and Li, F. (2010). SJTULTLAB: Chunk Based Method for Keyphrase Extraction. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 158–161.

Wilkins, V. (2012). *Understanding Loyalty and Motivation of Professional Sports Fans*. Master of science in sport and leisure service management, University of Nevada.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C. A., and Nevill-Manning, C. G. . (1999). KEA: Practical Automatic Keyphrase Extraction. In *Proceedings of the 4th ACM Conference on Digital Libraries*, pages 254–255.

Yang, D. and Powers, D. M. (2006). Verb Similarity on the Taxonomy of WordNet. In *Proceedings of Third International WordNet Conference (GWC-06)*, pages 121–128.

Yazdani, M. and Popescu-Belis, A. (2013). Computing Text Semantic Relatedness using the Contents and Links of a Hypertext Encyclopedia. *Artificial Intelligence*, 194:176–202.

Zajic, D., Dorr, B., and Schwartz, R. (2002). Automatic Headline Generation for Newspaper Stories. In *Workshop on Automatic Summarization*, pages 78–85.

Zesch, T. (2009). *Study of Semantic Relatedness of Words Using Collaboratively Constructed Semantic Resources*. PhD thesis, Darmstadt University of Technology.

Zesch, T. and Gurevych, I. (2009). Approximate Matching for Evaluating Keyphrase Extraction. In *Proceedings of the 7th International Conference on Recent Advances in Natural Language Processing*, pages 484–489.

Zesch, T., Müller, C., and Gurevych, I. (2008a). Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, pages 1646–1652.

Zesch, T., Müller, C., and Gurevych, I. (2008b). Using Wiktionary for Computing Semantic Relatedness. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 861–867, Chicago, IL, USA.

Zhang, W., Su, J., Lim, C., Wen, T., and Wang, T. (2010). Entity Linking Leveraging Automatically Generated Annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1290–1298.

Zheng, Z., Li, F., Huang, M., and Zhu, X. (2010). Learning to Link Entities with Knowledge Base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–491.

Wissenschaftlicher Werdegang des Verfassers[¶]

10/03 – 09/04	Fernstudium der Physik (Bachelor) an der Technischen Universität. Kaiserslautern.
10/04 – 10/07	Studium der Physik (Bachelor) mit Nebenfach Informatik an der Technischen Universität Darmstadt.
06/06 – 02/07	Studium der Physik (Auslandssemester) an der Pontifícia Universidade Católica do Rio de Janeiro, Brasilien.
10/07	Abschluss als Bachelor of Science. Bachelor-Thesis: “Bestimmung der Pion-Zerfallskonstante und des chiralen Kondensats aus der Struktur des Quark-Propagators” Gutachter: Prof. Dr. Christian Fischer
04/07 – 03/10	Studium der Physik (Master) mit Nebenfach Informatik an der Technischen Universität Darmstadt.
03/10	Abschluss als Master of Science. Master-Thesis: “Extracting the learner’s opinions - Bootstrapping extraction patterns for sentiment analysis” Gutachter: Prof. Dr. Iryna Gurevych
seit 05/10	Wissenschaftlicher Mitarbeiter am Ubiquitous Knowledge Processing Lab, Technische Universität Darmstadt.

[¶] Gemäß §20 Abs. 3 der Promotionsordnung der TU Darmstadt

Ehrenwörtliche Erklärung[¶]

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades Dr.-Ing mit dem Titel “Approaches to Automatic Text Structuring” selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Darmstadt, den 19. Dezember 2014

M.Sc. Nicolai Erbs

[¶] Gemäß §9 Abs. 1 der Promotionsordnung der TU Darmstadt

Publication Record

We have previously published the main contributions of this thesis in peer-reviewed conference or workshop proceedings of major events in natural language processing and related fields, such as ACL, RANLP, ICSC and D-Lib. The chapters which build upon these publications are indicated accordingly.

Nicolai Erbs, Iryna Gurevych and Torsten Zesch: 'Bringing Order to Digital Libraries: From Keyphrase Extraction to Index Term Assignment', in: *D-Lib Magazine*, vol. 19, no. 9/10, pp. 1-16, September 2013. (Chapter 3)

Nicolai Erbs, Pedro Bispo Santos, Iryna Gurevych and Torsten Zesch: 'DKPro Keyphrases: 'Flexible and Reusable Keyphrase Extraction Experiments', in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. System Demonstrations*, pp. 31-36, Baltimore, MD, USA, June 2014. (Chapter 3 and Appendix A)

Jinseok Nam, Christian Kirschner, Zheng Ma, **Nicolai Erbs**, Susanne Neumann, Daniela Oelke, Steffen Remus, Chris Biemann, Judith Eckle-Kohler, Johannes Fürnkranz, Iryna Gurevych, Marc Rittberger, and Karsten Weihe: 'Knowledge Discovery in Scientific Literature', in: *Proceedings of the 12th Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2014)*, pp. 66-76, Hildesheim, Germany, October 2014. (Chapter 3)

Nicolai Erbs, Iryna Gurevych and Torsten Zesch: 'Hierarchy Identification for Automatically Generating Table-of-Contents', in: *Proceedings of 9th Conference on Recent Advances in Natural Language Processing (RANLP 2013)*, pp. 252-260, Hissar, Bulgaria, September 2013. (Chapter 4)

Nicolai Erbs, Torsten Zesch and Iryna Gurevych: 'Link Discovery: A Comprehensive Analysis', in: *Proceedings of the 5th IEEE International Conference on Semantic Computing (IEEE-ICSC)*, pp. 83-86, Palo Alto, CA, USA, July 2011. (Chapter 5)

Nicolai Erbs, Eneko Agirre, Aitor Soroa, Ander Barrena, Ugaitz Etxebarria, Iryna Gurevych, and Torsten Zesch. 'UKP-UBC Entity Linking at TAC-KBP', in: *Proceedings of the 5th Text Analysis Conference, Workshop Papers*, Online Proceedings, Gaithersburg, MD, USA, November 2012. (Chapter 5)

Tristan Miller, **Nicolai Erbs**, Hans-Peter Zorn, Torsten Zesch and Iryna Gurevych: 'DKPro WSD: A Generalized UIMA-based Framework for Word Sense Disambiguation', in: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. System Demonstrations*, pp. 37-42, Sofia, Bulgaria, August 2013. (Chapter 5)

Nicolai Erbs, Iryna Gurevych and Torsten Zesch: 'Sense and Similarity: A Study of Sense-level Similarity Measures', in: *Proceedings of the 3rd Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pp. 30-39, Dublin, Ireland, August 2014. (Chapter 5)

Daniel Bär, **Nicolai Erbs**, Torsten Zesch and Iryna Gurevych: 'Wikulu: An Extensible Architecture for Integrating Natural Language Processing Techniques With Wikis', in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics. System Demonstrations*, pp. 74-79, Portland, OR, USA, June 2011. (Chapter 6)

Wolfgang Stille, **Nicolai Erbs**, Torsten Zesch, Iryna Gurevych, and Karsten Weihe: 'Aufbereitung und Strukturierung von Information mittels automatischer Sprachverarbeitung', in: *Proceedings of KnowTech*, pp. 199-208, Bad Homburg, Germany, September 2011. (Chapter 6)

Nicolai Erbs, Daniel Bär, Iryna Gurevych, Torsten Zesch: 'First Aid for Information Chaos in Wikis: Collaborative Information Management Enhanced Through Language Technology', in: *Information und Wissen: global, sozial und frei? : Proceedings des 12. Internationalen Symposiums für Informationswissenschaften*, pp. 501-502, Hildesheim, Germany, 2011. (Chapter 6)