Explorative Analysis of Structure and Semantics in Topic-Coherent News Articles.

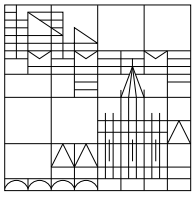
Masterarbeit

zur Erlangung des akademischen Grades eines
Master of Science (M.Sc.)

vorgelegt von

Michael Behrisch

an der



Universität
Konstanz

Fachbereich für Informatik und Informationswissenschaft
Master-Studiengang Information Engineering

1. Gutachter: Prof. Dr. Daniel Keim
2. Gutachter: Jun.-Prof. Dr. Tobias Schreck

Konstanz, 29.09.2011

Behrisch, Michael:
Explorative Analysis of Structure and Semantics in Topic-Coherent News Articles.
Masterarbeit, Universität Konstanz, 2011.

Danksagung

Ich widme diese Arbeit einer sehr wichtigen Person in meinem Leben, die mich unglaublich unterstützt, beflügelt und in guten –wie in schlechten– Zeiten für mich da ist: Susanne. Danke, dass du zu mir hältst, mir von deiner unglaublichen Power abgibst und mich zudem immer schimpfst, wenn ich einmal wieder vor lauter Arbeit das Essen vergessen habe. Gerade die letzten Wochen und Monate haben mir gezeigt, dass du mir eine unendlich wertvolle Stütze bist und mir mein Leben versüßt. Ich freue mich auf alles Kleine und Große was noch vor uns liegt.

Ein weiterer großer Dank gilt meiner Familie, die mit genauso großem Elan und Freude dem Ende meiner Studienzeit entgegen sieht. Danke für jede eurer Unterstützungen, ob sie nun verbal oder materiell waren. Danke Mama, dass du immer zuhörst, ich auf dich zählen kann und du immer an mich geglaubt hast. Danke Papa für dein Interesse, dein Blick für die Zukunft und deine Hilfsbereitschaft in allen Lebenslagen. Danke Andrea und Basti, auf deren Meinung ich immer viel gegeben habe. Auch danke ich meiner Oma, die mit 90 Jahren zwar noch nie den Sinn und Zweck eines Computers verstanden hat, aber immer Interesse gezeigt hat.

Ein weiterer Dank gilt meinem Betreuer, Miloš Krstajić, der mir auf der einen Seite freie Hand bei der Auswahl des Masterarbeits-Themas und jeder Entscheidung auf dem Weg dahin gelassen hat, und mir im gleichen Maße mit Rat und Tat zu Seite stand.

Einen letzten Gruß möchte ich meinen Freunden aussprechen. Danke, dass ich so tolle und liebenswerte Freunde hier in Konstanz finden durfte und dass die festen Bausteine meines Darmstädter Lebens geblieben sind.

Zusammenfassung

Nachrichtendaten sind omnipresent in unserer Gesellschaft und fungieren als ein wichtiger Bestandteil im Meinungsbildungsprozess für weite Teile unseres täglichen Lebens. Politische, ökonomische und sozio-kulturelle Entscheidungen werden oftmals auf Basis der Nachrichteninformationen gefällt. Aus diesem Grund kristallisiert sich – auf der einen Seite– mehr und mehr die Notwendigkeit heraus auf dem neustem Stand bleiben zu müssen und –auf der anderen Seite– eine kritische Sichtweise auf die schier überwältigendende Masse an Daten einzunehmen. Doch gerade der Vorstoß des Internets in die Domäne der Nachrichten-Distribution lässt die Nutzer in einem Meer an widersprüchlichen, redundanten oder Kontext entfremdeten Informationen ertrinken. Gerade dies erschwert es wohl informierte Entscheidung treffen zu können.
Diese Masterarbeit hat das Hauptziel Methodiken und Techniken zu entwickeln, die den Nutzer in einem investigativen Leseprozess unterstützen. Aus diesem Grund präsentiert die Arbeit eine abstrakte Nachrichtendaten Explorations-Pipeline, mit deren Hilfe die essentiellen Punkte einer profunden Datenanalyse erklärt und in den Kontext gestellt werden. In den Hauptkapiteln wird zum Einen ein allumfassender Überblick über das Feld gewährt und zum Anderen die wichtigsten Techniken detailiert vorgestellt. Wichtiger noch ist, dass diese Arbeit ein voll funktionstüchtiges Nachrichtendaten Explorationssystem –namens *NewsGuide*– präsentiert. NewsGuide wurde im Rahmen der Masterarbeit entwickelt und hat den Fokus neue, als auch etablierte, Herangehensweisen zu kombinieren und mit deren Hilfe den Leseprozess effizienter zu gestalten. Hierfür ermöglicht das System sowohl eine zeit-, als auch kontext-abhängige Exploration der Nachrichtendaten. Weiterhin soll NewsGuide den Leser in seinem Leseprozess steuern und unterstützen. Aus diesem Grund wurde eine “Recommendation Engine” entwickelt, die ihre Entscheidung nicht nur auf die wohlbekannten textuellen Ähnlichkeitsmaße fußt, sondern vielmehr semantische Aspekte einfließen lässt. In diesem Kontext wird das “Bag-of-Synset Modell” vorgestellt, eruiert und evaluiert. NewsGuide zeigt seine Anwendbarkeit in einer Vielzahl von Explorationsteilfeldern, wie zum Beispiel der extraktiven Multi-Dokument Textzusammenfassung, Inter-/Intra- Nachrichtentext Vergleich und in einem neuen Analyseschritt, der Haupt-Handlungsstrang Extraktion.

Abstract

News data is omnipresent in our society and functions as an important decision influencing factor for a wide range of fields. Political, economical, and socio-cultural decisions are often based on news information. A range of consequences can be derived from this fact. Thus, it becomes more and more vital for the user to stay up-to-date and –on the other hand– to develop a more critical view onto the sheer overwhelming amount of news data from multiple news sources. The latter case is even put to the extreme due to the Internet's advance into the field of news distribution. Particularly here the readers are confound with contradictory, redundant and context-diverting information, which hinders them in reaching an educated decision.

This master thesis has the overall goal to develop means and techniques for the investigative exploration of news topic clusters, which consist of articles from multiple news sources. Therefore, a news topic exploration pipeline, which facilitates a user's examination, is presented. A comprehensive overview of text similarity measures, text document clustering techniques, text summarization and related work in the area of text mining and visualization is given. Moreover, the thesis presents *NewsGuide*, a fully functional news data exploration system, which allows for an exploration of news topics on the textual, structural and semantic layer. It incorporates new methods for guiding the user in an efficient reading process. The developed article suggestion technique does not only depend on the well-known textual similarity measures, but rather leverages the process through semantics. Additionally, a topically-enhanced text summarization technique is presented. In this context, the Bag-of-Synset textual model is presented, investigated and enhanced with a new semantic relatedness measure. NewsGuide demonstrates its applicability in several news exploration subfields, such as multi-document extractive text summarization, inter-/intra- document news comparison and a novel sophisticated analysis step: main storyline-extraction.

Table of Contents

Table of Figures

Chapter 1

Introduction

1.1 Motivation

News data has advanced to one of the most prominent decision influencing factors in our data-driven society. While in its early days the news landscape was dominated by print media, today's news distribution is highly affected by the fast changing media Internet. According to a research study the Internet has surpassed newspapers as a primary way for Americans to get news [Pew11b], [Pew11c].

But news are not only raw information sources. They also have to be seen as a powerful influencer for socio-cultural values and political decisions. As an example, in January 2011 began the Egyptian mass demonstrations against their political leadership. Primarily, Internet-driven information sources, such as social networks (i.e. Facebook [Fac11] and Twitter [Twi11]), or various blogs and other news organizations encouraged a political consciousness and awareness within Egypt and also worldwide.

But, news do not only have a socio-cultural value. From an economical point-of-view, news data influences stock markets, drives acquisition decisions and leads to market cooperations. Thus, it has the power to drive multi-million dollar decisions and affects entire economy sectors.

The Internet, as the primary platform for accessing news, brings vast improvements for both, the publisher and the consumer. It not only allows content providers to publish information easily and efficiently, but it also enables users to find a broad mass of information, attitudes, and aspects for a specific topic. However, while in earlier days the reader subscribed one, two, or at most three magazines and newspapers, he is now confronted with a myriad of news sources. Albeit, news categories, search abilities and specialized content feeds enable the user to stay up-to-date with his focus topic, the vast amount of content sites brings an overwhelming mass of news data.

Even more, the already overwhelming amount of news content is further enlarged by means of the Web 2.0 sphere. The Web 2.0 movement sees the Internet not as a

static media, but as an [...] "interactive experience in the form of blogs, wikis, forums, etc., and plays a more important role than simply accessing information" [Dic11]. User-generated content, such as comments, reviews and ratings are one of the outstanding achievements of the Web 2.0 sphere. Not only do they cover entirely subjective opinions, but they also open up other individual point-of-views or give an insight into the crowds' knowledge. Here, subjectivity and objectivity are intertwined and enable the user to interact with the web content. Nevertheless, while the interactive web allows a much more comprehensive information exchange in comparison to print media, it also forces the user to search through a large variety of news sites, all with different opinions and thoughts on a topic. This fact is further strengthened by the Internet's predominant opinion that information is freely and ubiquitously available.

1.1.1 The Internet News Landscape

Despite its undoubtedly decision influencing factor, the Internet's news data landscape is diverse. As Figure 1.1 depicts, the news value chain can primarily be subdivided into four main stages: Content creation, advertising, manufacturing and distribution. While, from this project's point-of-view the news distribution process is the most important one, the other steps also have to be taken into account in order to give a comprehensive insight.

Content Creation
Content creation in the journalistic field is an elaborate process based on background research, investigation and other activities. Its main goal is to produce textual-, video- or audio content, which can be used for later journalistic practices. Journalistic content is mainly produced by three branches of authors.

1. News and Photo Agencies: News and photo agencies are organizations of journalists with the goal to produce and sell news content to news organizations. They are the largest provider of hard news facts, which either go in directly or serve as background information to feature news articles. Well-known media company affiliated news agencies are, for example, the Press Association, Agence France-Presse (Afp) or All-Headline-News (Ahn). News agencies with different intentions exist, as well. Redistributors, such as the Associated Press (Ap) or the Deutsche Presse Agentur (Dpa), aggregate, filter and potentially reinvestigate/enhance other news agencies' and/or journalists' news stories. As the final product they redistribute local or global news stories. Further examples are Xinhua (China) or Itar-Tass (Russia), which are governmental-funded news agencies distributing nationally interesting and governmental news globally and within a country's frontiers.

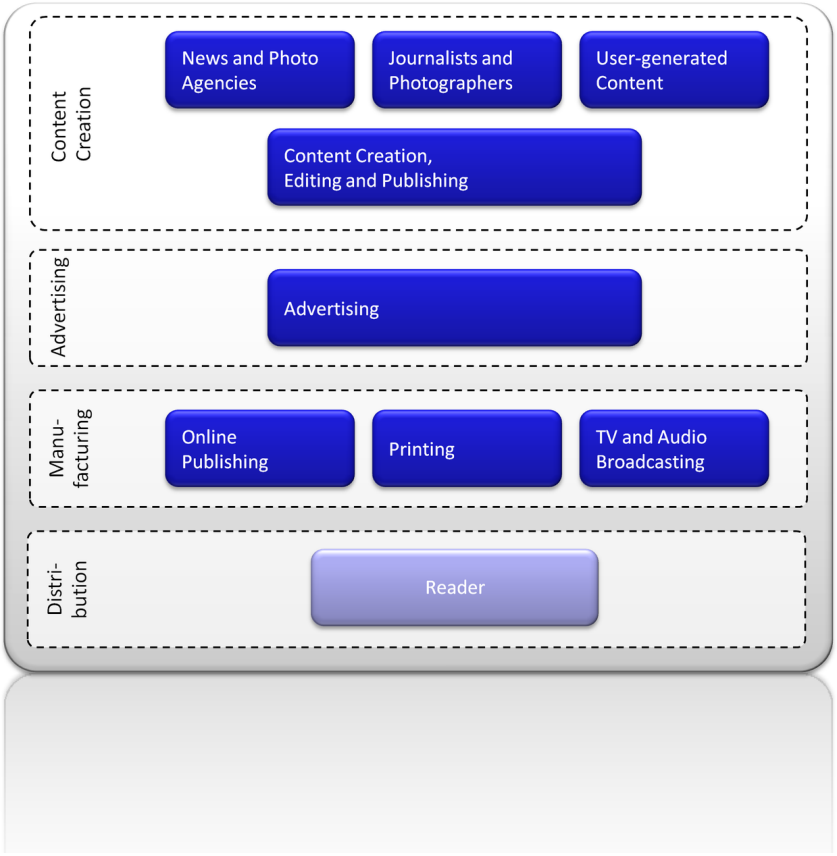


Figure 1.1: The News Landscape.

2. Journalists and Photographers: Whenever journalists or photographers are not associated with a news agency they are called free journalists. This group is well known for their impartiality. Free journalists are mostly freelancers, who work intermittently for news agencies or news organizations.

3. User-generated Content: Another new and outstanding source of information content is a product of the Web 2.0 Internet development. User-generated content allows the dissemination of reviews, comments or ratings that are free of journalistic conventions. Accordingly, it enables users to re-frame the public opinion about specific topics. Particularly, customer reviews are widespread and an often used user-generated content source.
   Another –yet extreme– example of user-generated content is real-time citizen journalism:

   > *There's a plane in the Hudson. I'm on the ferry going to pick up the people. Crazy.* 9:36 AM Jan 16th from jkrums
   > http://twitpic.com/135xa [Kru09]

This kind of *citizen journalism* brings a new streaming into the journalistic domain. It allows every Internet-connected user to publish content and gives means for a real-time coverage of events.

Advertising
The advertising business is the primary revenue generator for the news landscape and is followed by subscription-based revenues. According to a 2010 published research study [OEC11] the global print newspaper publishing market derived about 57% of its revenues from advertising and about 43% from newspaper sales. The global newspaper publishing market (defined as online and offline circulation and advertising revenues of traditional newspaper publishers) is estimated at USD 164 billion in 2009 [Pri09].

Manufacturing
After the informational content has been created the manufacturing process begins. Here the news organizations, or news providers, transform the "raw" information or hard facts into a presentable news story. The manufacturing workflow includes copy-editing, editing, re-arranging and graphical work (arranging articles, pictures, design) and finally the creation of a fully digital version ready for the distribution process.
Newspaper organizations have manifold appearances. The most prominent examples are newspapers, magazines, radio- or television broadcasters. Since the focus of this work is on textual data, audio- or video representations of news are omitted.

1.1.2 News Content Distribution

After the news story has been created, it is either brought into a printable version or presented through a news organization's online platform. Most large newspapers and magazines have an online platform for accessing their information. Figure 1.2 shows such an online portal [Tim11].

By means of an online news portal users have the option to browse through the news content and catch up with the latest developments in a specific field. Generally, only one topic is presented with a headline and later updates replace the link on the specific news page. Hence, the update is generally not distinguishable from the previous version. As a direct result, the reader is left without a trace of the story's development, whenever a story is not marked as "updated". Furthermore, a news organization's site does not reveal the news structure itself, because XML-like markup is usually not transmitted. This makes web-crawling generally a difficult and error-prune task.

As another way of digital content distribution, Rss (Rdf Site Summary, often dubbed Really Simple Syndication) has to be taken into consideration. Rss is a family of web feed

Figure 1.2: The New York Times Online Portal [Tim11].

formats used to publish frequently updated works –such as blog entries, news headlines, audio, and video– in a standardized format. Figure 1.3 depicts a Rss Xml representation.

Rss' main advantage is that content can be distributed in a structured and machine-readable fashion. Thus, important primary information (e.g. the title, the publication and modification dates, and description/full-text) can be easily identified and metadata can be used to enhance the semantic meaning of the Rss file. For example, the *pubDate* tag allows answering the question, which news article reported first about a topic.
As a further advantage, Rss feeds allow users to personalize their daily news selection, either by subscribing to different feeds from a variety of news organizations or by choosing a news organization's branch feed (i.e. politics, technology, or sports, etc.).

In the case of a far-reaching analysis of news data –as in this project– Rss can be seen as a key enabler for content aggregation. It allows the structured processing of large

Figure 1.3: Rss Xml Representation [Wik11].

amounts of news items from different news organizations and -agencies. Only through this news aggregation step, questions like "Which news topics exist?", or "Who wrote first about the topic?" can be answered in an appropriate manner.

Figure 1.4 presents a news aggregator online portal [Goo11]. Its main advantage is that the news stream of (nearly) all major news organizations and agencies is regularly fetched, processed and presented in a standard interface to the user. Other news data aggregators exist, as well. The most prominent ones are Google News [Goo11], Yahoo News [Yah11], Ask.com News [Ask11] or BigNews.biz [Big11].
Their common features are the fetching of the news items from a large list of content providers and their assignment to a news story. In more technical terms, they iteratively assign each new incoming news item to a cluster of topic-coherent news. Additionally, they attach topic related photographs, audio- or video content and provide basic statistics for the clusters. However, a significantly improved and augmented news preprocessing is possible. For example, the Europe Media Monitor, or short EMM [Mon11c], features for example news data clustering, named entity extraction, approximate name matching, multi-language alignment, multi-media information enhancement, and geo-location information. Due to its sophisticated standing in the news exploration domain, a further enhanced description of the EMM system can be found in Section 2.1. Here also a detailed explanation of the mentioned techniques will be given.
EMM has to be seen as the most sophisticated news data aggregator in comparison to the latter mentioned. Not only that it provides an user-friendly interface, but it also

Figure 1.4: The Google News Aggregator [Goo11].

enables an exploration of the news landscape. EMM NewsExplorer [Mon11c] should even be seen as a visual analytics tool for the news domain, which still can be enhanced with more sophisticated methods and approaches. The following Chapter 1.1.3 "News Usage Scenarios" will reveal some of the shortcomings and extension-points, which led to the motivation of this master thesis.

1.1.3 News Usage Scenarios

The way news are consumed has drastically changed in the last decade. While in earlier days the users had a limited and news organization-centric view, they are now fraught with a myriad of content sources. Hence, the user has to spend increasingly effort to filter the important and trustworthy articles from a large pool of news. Generally, this determines a user's news reading behavior. According to user studies, e.g. [GS91]

or [HHBL03], news consumers can be categorized into two user groups with different information intentions:

News Scanning Readers
This group of news consumers spends little to no effort on reading news on the Internet. Usually, they scan through a large amount of headlines and abstracts and choose only the personally most interesting news. Eye-tracking studies of (online-) newspaper reading, e.g. such as in [GS91] or [HHBL03], have found that readers regularly skim by scanning graphics, headlines, and initial paragraphs before intermittently stopping to read an article. As a direct result this user group prefers news story abstracts, recommendations and eye-catching headlines.

Deep Dive News Reader
This group of news consumers spends a lot of time reading news articles on the Internet. Their intention is either intrinsically or extrinsically motivated. Intrinsically driven users feel a deep connection to one or more specific fields of interest. On the other hand, extrinsically motivated users have to have a deep knowledge in a specific field. Otherwise, they would lose track and thus information, which hinders i.e. their professional advancement. This group distinguishes from the next group by their amount of previous knowledge. Deep dive news readers develop their knowledge from the scratch. They usually prefer full-fledged news stories with a lot of background information [GS91]. Nevertheless, this user group would appreciate to keep track of how much knowledge they already got and how much information is not accessed yet.

Up-to-date News Reader
Up-to-date news readers spend a medium amount of time to keep track with the latest developments in a field. Generally, this group has already basic or enhanced background knowledge and spends time to increase this information repository. Albeit, up-to-date readers are a subgroup of the deep dive user group this group prefers a fast, accurate and short news update [GS91].

Mobile Internet News Users
Another development has also to be taken into account. The mobile Internet –once again– changes the news' appearance and the way news sites are approached. 47% of the American adults use their cell phones and tablets to get local news and information [Pew11a]. However, the reading of news on small screens also affects the usage behavior. While larger screens and more comfortable input mechanisms allow longer and full-featured news stories, the user today wants to have fast and on-the-fly insight into unknown topics while he/she i.e. waits for the bus. Summaries, tag clouds, or personal-interest alarms are major enhancements to get grip onto the textual data explosion.

1.2 Primary Objectives

Based on the results of their study, Garcia and Stark [GS91] define newspaper design as the challenge "to give readers material that is worthy of their scan and that makes them stop scanning and start reading." This statement can be enlarged to the primary objectives of the news exploration domain. In this project it is the primary goal to support the users by finding interesting and worthy headlines, which make them stop scanning through the never-ending news article stream and makes them start reading on a specific topic.

Figure 1.5: Primary Objectives; Images courtesy [Goo11] and [Mon11b].

Figure 1.5 illustrates the news domain's primary problem and thus derives the project's primary objective. In the Internet news landscape the user is confronted with an overwhelming amount of textual news data. News data aggregators, such as the ones introduced in Chapter 1.1.2 usually present a large amount of topic-interrelated news articles. While this gives –on the one hand– the advantage to have all information in one place, it still leaves the user without any knowledge about the overlap within the news cluster. However, most standard methods, such as the ones which will be presented in Chapter 3.2 are able to measure the textual overlap, in the sense of using the same words, they fail to show the varying aspects or semantical overlap. Most generally, every news article in a topic cluster has the same theme (otherwise the news clustering process has failed). Accordingly, the storyline of a full-fledged news article should be consistent with the storyline of all other news articles in the cluster. Despite this fact, the news articles vary greatly in the mentioned aspects or sub-stories.
From an user-perspective it would be sufficient to be recommended to one news article, which covers the main aspects, and then be left with the choice to read further aspects.

With this recommendation system the user would have to read less news and still get the same insight. As a direct result, users would spend less time reading redundant news facts.
However while less reading attracts most news reader user groups, it does not seem to be the primary goal for the other user groups. This is due to the fact that it does not allow to answer news evolution questions. Here, the user asks questions like "Which news article reported first on a story?" or "Is there a news article, which influenced the story progression significantly?".

1.2.1 Central Questions of News Data Analysis

The central questions of news data analysis can be subsumed into the following problems and their correspondent intentions:

- Who wrote first about the News?: The first report on a news story can be seen as the starting point of the following news topic stream. Consequently, it is interesting to find the first article, extract its storyline, argumentation and aspects in order to compare them to the upcoming articles.

- Who changes the News Content?: Whenever an influencing news article forms the topic's shape (such as the initial news article), further authors want to contrast the beliefs with their attitudes and aspects. As a result, it is interesting to ask Who influences the topic and hypothesize Why the author wants to bring his opinion into the spotlight.

- Who produces and who influences the News (Journalistic Level)?: As already mentioned before, most authors have the intention to develop a new story. On the other hand, the broad mass of news –especially in the Internet– did not come up the the topic on their own, but rather enhance or re-shape (influence the story stream) without having a large impact. Hence, it will be interesting to find out which agencies do a lot of background research and who are the "copycats".

- What was changed (Wording, Structure, Information)?: In addition to the latter mentioned point it is interesting to investigate whether news change entirely on a structural level (without information gain/loss) or reveal an information-delta to a reference article. Visual analytic methods can help to assess these structural and information differences/similarities.

- When was text modified?: Time is an important factor when assessing the controversy of a story. Thus, asking the question When was the text modified helps to explore the topic's novelty and salience.

- Can we derive Patterns?: Patterns are (non-)obvious trends which re-occur in the entire news cluster. In this context, it is interesting to see –for example– whether one news agency regularly copies text from another news agency or if several news articles "borrow" outstanding sentences for their own information (and reveal whether they cite the copying accurrately).

1.2.2 Project Objectives

The goal of the project is to give means to answer some of the primary questions from the latter Chapter 1.2.1. When we investigate the central questions more closely one can come to the conclusion that their focus could be split into a textual- and semantic information focus. The textual information change analysis tries to answer questions regarding the structural and textual appearance of a news article. On the other hand, the semantic information questions approach the fundamental facts that a news article tries to highlight.
Since the project objectives changed during the project runtime from the first to the second focus, the approaches for the first and second iteration changed likewise:

Project Phase 1

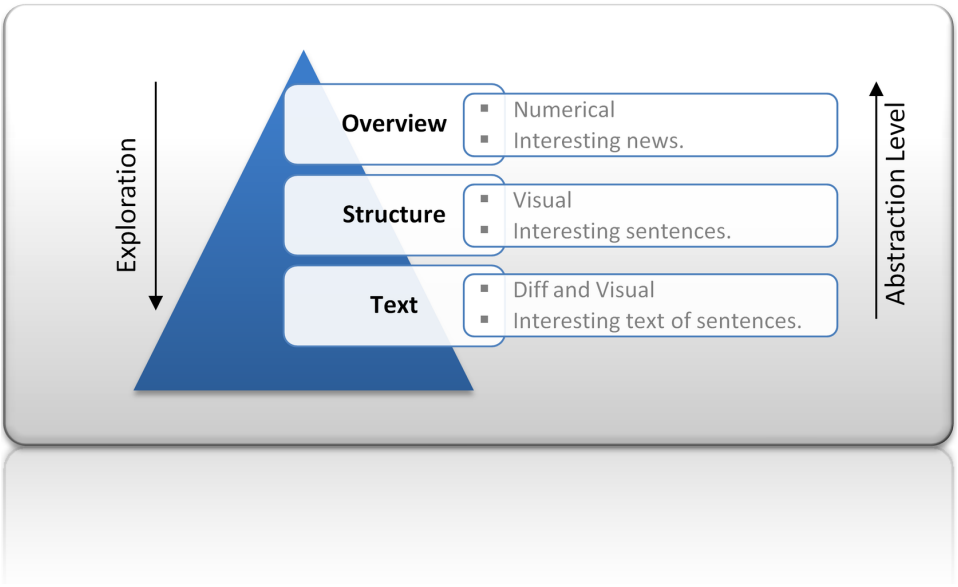


Figure 1.6: Project Objectives Phase 1.

Project phase 1 had the goal to answer central questions regarding the structural news evolution over time. With the assumption that the first news article about a

specific topic defines its primary direction, we investigated e.g. text-passage recurrence questions and tried to extract exploratively other non-obvious patterns with visual analytic means. As one exemplary result, we found that several news agencies publish their content in regular time periods to the news aggregators without any text modification. This leads to the assumption that the most widespread news crawlers (e.g. Google News or Yahoo News) are taking the update-frequency of a news item into account to evaluate its importance. With a frequent update of the news site these rating mechanisms can be tricked and result in higher click-through rates for the news organization's web site.

Project Phase 2

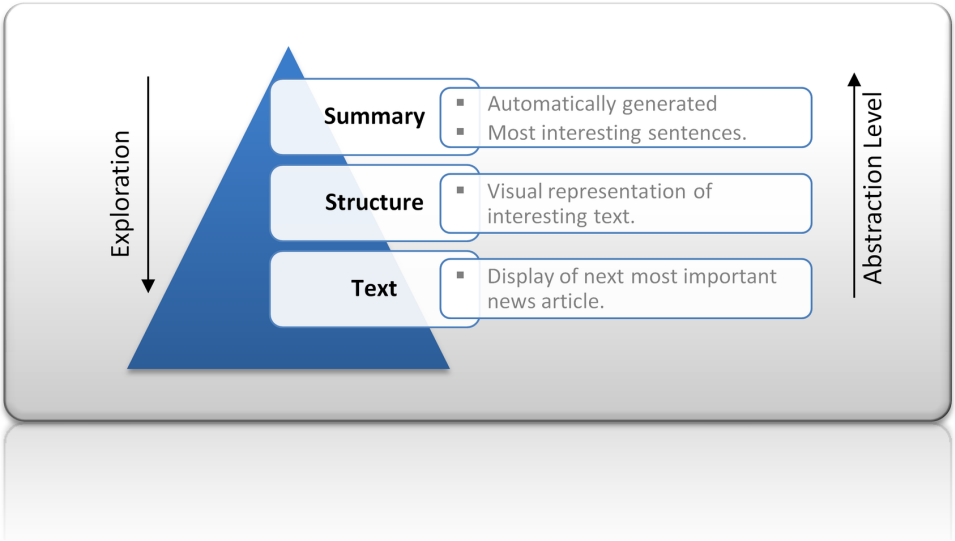


Figure 1.7: Project Objectives Phase 2.

Project phase 2 had the goal to support the user in his news reading process. Its objective was to emphasize and measure the semantic differences between news articles. Whenever readers try to approach a new topic they are inevitably confronted with a large amount of semantic overlap within the news cluster. Accordingly, the majority of the mentioned aspects will be read more than once. Conversely, the readers generally have no idea of how much information they already gained during the reading process. Thus, the second project phase tried to establish a recommendation engine that iteratively guides the users through the news cluster exploration.

1.3 Overview

The rest of this master thesis is structured as follows. First, Section 2 "*Related Work*" will introduce the most important available news data exploration systems and gives an insight into their approaches towards the topic. Second, in the first main part (Section 3 "*News Data Exploration Techniques*") all important techniques of a news exploration system will be described in detail. It will highlight natural language processing techniques (i.e. named entity extraction, text summarization), as well as data mining mechanisms (i.e. text data clustering, textual similarity measures) and user-interfaces/visualization approaches (i.e. text data visualizations, relevance feedback). Third, in the following primary chapter of this thesis (Section 4 "*NewsGuide - News Exploration System*") the outcome of the master project "News Data Exploration" will showcased and evaluated in comparison to the approaches in this field. Finally, Section 5.1 "*Conclusion and Future Work*" will conclude the thesis with comments and an outlook into the future of news exploration systems.

Chapter 2

Related Work

This chapter will describe some of the related approaches in the field of news data exploration. It will give an exemplary insight into promising or sophisticated implementation approaches for the domain. It is noteworthy that this section should neither function as an exhaustive list of research projects, nor as a complete enumeration of research groups within the field. Rather, it should give a holistic overview about the current state of research and present interesting approaches towards the problems occurring in the targeted problem domain. As a result of this, it will emphasize only full-fledged news exploration systems and not encapsulated techniques for processing textual/news data. A detailed description for most of the developed techniques will be given in-depth in the sub-chapters of the Section 3 "*News Data Exploration Techniques*".

2.1 EMM NewsExplorer

The European Media Monitor, short EMM, is a joint research project of the DG-JRC (Directorates-General Joint Research Centre) and the DG-Press (Directorates-General Press Service). The Joint Research Centre, short JRC, is the European Union's scientific and technical research laboratory and an integral constituent of the European Commission. JRC has developed a number of news aggregation and analysis systems with the primary objective to support EU institutions and member state organizations. Currently, three web portals (EMM NewsBrief [Mon11b], EMM NewsExplorer [Mon11c] and EMM MedISys [Mon11a]) are publicly accessible. Figure 2.1 (adapted from [BRS06]) shows the EMM processing chain and the interrelations between the news sites.

EMM enhances in its processing chain aggregated news articles with metadata. This metadata is automatically generated by software algorithms without any human intervention.

Figure 2.2 shows the front page of the NewsBrief portal. It consists of the latest

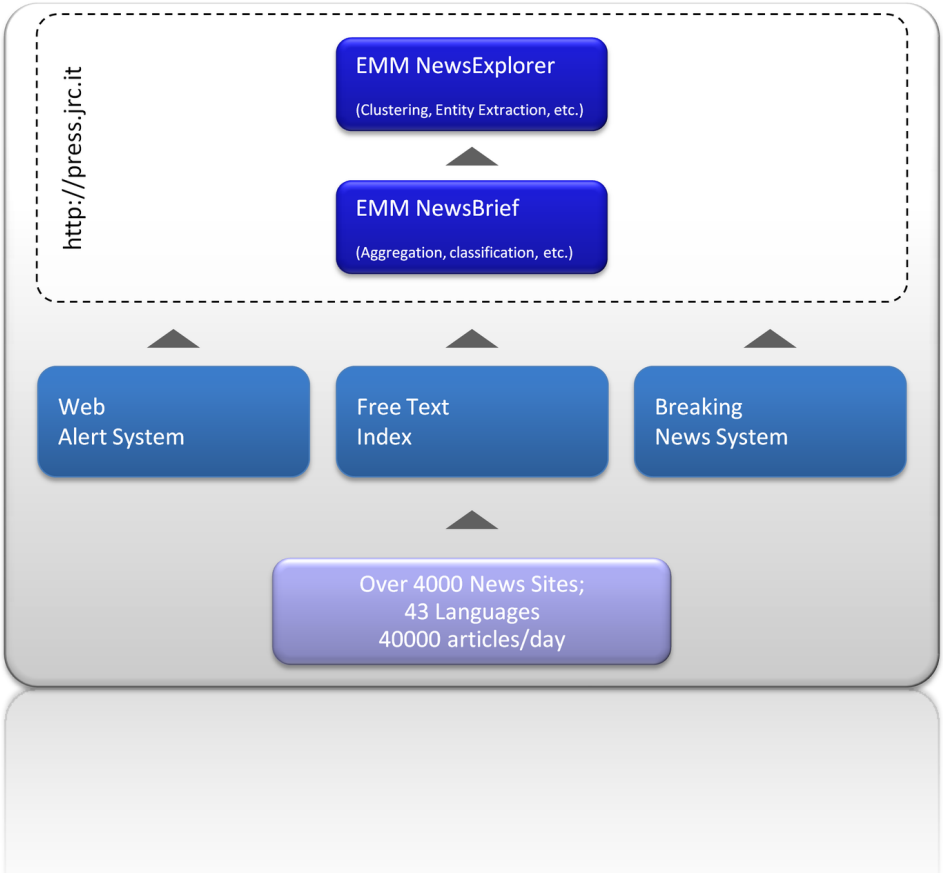


Figure 2.1: European Media Monitor Processing Chain; Adapted from [BRS06].

top stories as detected by EMM's topic detection and clustering techniques (also called "Breaking News System"). Regarding the functionality EMM NewsBrief is comparable to Google News or Yahoo News. It gathers and excerpts current news stories. The user can focus news by various top stories, 24 hours overview, by various themes, by region of the world or country.

The other publicly available news analysis tool, called EMM NewsExplorer, goes even one step further and features the following characteristics (information adapted from [Mon11c]):

- Clustering: Clusters all news articles of the day, separately for each of the 43 languages, into groups of related articles. The EMM NewsExplorer uses a hierarchical clustering algorithm, which will be described in Section 3.1.1.
- Named Entity Extraction: Identifies names of people, places and organizations for each cluster. The related techniques for this functionality can be found

Figure 2.2: European Media Monitor: NewsBrief Portal 2.1.

in Section 3.5.2.

- Approximate Name Matching: Applies approximate name matching techniques to all names found in the same cluster, in order to identify which name variants may belong to the same person. A high matching degree will be achieved if an entity's name is similar to another entity's name. Accordingly, a textual similarity measure has to be defined, which is not too restrictive, but leaves no room for misunderstandings. A selection of textual similarity measures can be found in Section 3.2.

- Multi-Language Alignment: Links the monolingual clusters with the related clusters in the other languages.

- Cluster Representative: Identifies the most typical article of each cluster and uses its title for the cluster. A review of techniques for finding a cluster representative can be found in Section 3.3.

- Pattern Finding: Stores the extracted information in a database, which gives the chance to apply advanced machine-learning algorithms for (person-related, topic-related and location-related) time-series data.
- Information Enhancement: Occasionally, EMM searches automatically the Wikipedia online encyclopedia for images and for further multilingual name variants to enhance the article with more interesting facts.
- Geo-local Information: Adds geo tags to each news article depicting either the news agency's/organization's location or the extracted places.

Figure 2.3 shows a screenshot of the NewsExplorer online portal:

Figure 2.3: European Media Monitor: NewsExplorer Portal [Mon11c].

Alike, EMM NewsBrief it allows to access a myriad of information in a well-known visual interface. However, more importantly is EMM's external processing capabilities.

EMM publishes their metadata enhancements in the XML format for affiliated research organizations (as for example the University of Konstanz). The results can be used for further processing/analysis. Listing 2.1 shows the output, which can function as a starting point for further processing steps.

Algorithm 2.1: EMM XML Output

```
<item emm:id="bbc-78ee9c05b6c74cf842c409f7c3467c1a">
  <link>http://news.bbc.co.uk/2/hi/asia-pacific/8353451.stm
  </link>
  <pubDate>2009-11-10T19:19+0100</pubDate>
  <source url="http://news.bbc.co.uk"
    country="GB"
    rank="1">bbc</source>
  <iso:language>en</iso:language>
  <category emm:rank="1"
    emm:score="18"
    emm:trigger="warned[2]; deadly[2]; war[1]; fire[2];
    incidents[1]; warns[1]; protesting[1]; nuclear[3];
    warship[1]; weapons[3]; battles[1]; armed[2]; ">
      Security</category>
  <category emm:rank="1"
    emm:score="44"
    emm:trigger="North Korea[5]; Pyongyang[1]; ">
      NorthKorea</category>
  <emm:entity id="1510"
    type="p"
    count="1"
    pos="469"
    name="Barack Obama">Barack Obama</emm:entity>
  <emm:georss name="South Korea"
    id="192"
    lat="37.5424"
    lon="126.935"
    count="1"
    wordpos="23"
    class="0">South Korea</emm:georss>
</item>
```

2.2 Lydia

The Lydia project is the New York State University's natural language processing project with the goal to analyze information from various news sources. The project extracts predefined classes of facts and relations from curated text sources. Most prominently, Lydia's focus is set on determining temporal and spatial distributions of entities. It tries to answer questions, like who is being talked about, by whom, when, and where?
Lydia uses several natural language processing techniques to identify references to interesting people, places, or things and studies their relationship (they call it juxtapositions). The blog diagram in Figure 2.4 (left) shows the Lydia project's processing pipeline.

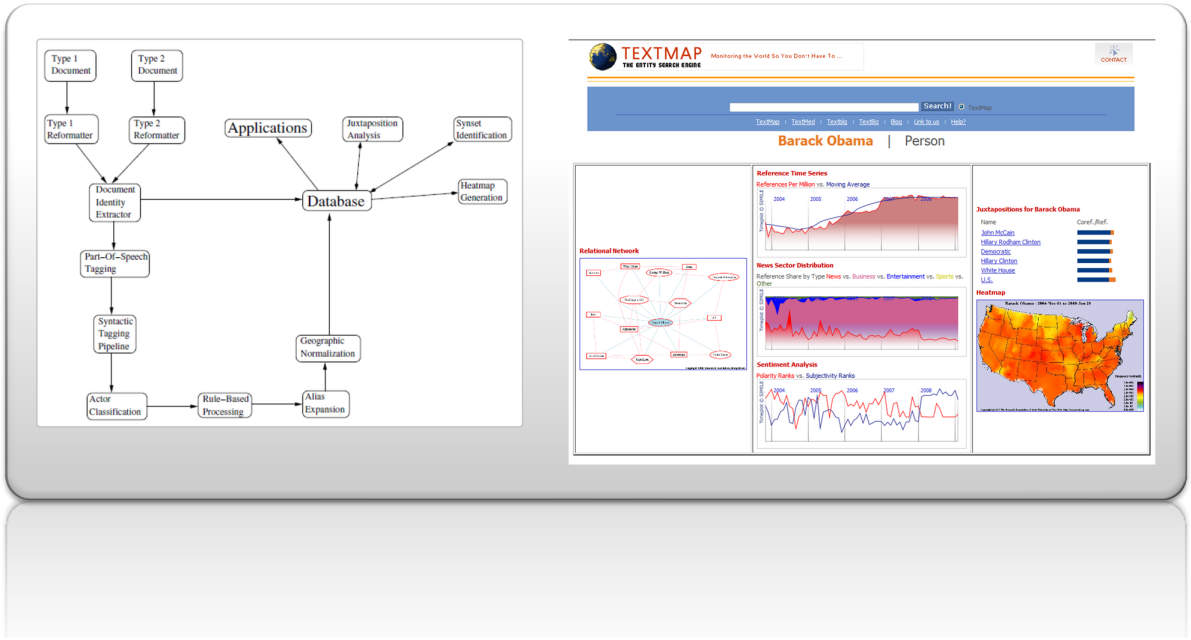


Figure 2.4: Lydia Processing Chain (left) [LKS05] and Web Front End (Right) [Tex11b].

The pipeline's major analysis phases can be subsumed under the following enumeration, which is adapted from [LKS05]:

- Named Entity Recognition: Identifies entities (people, places, companies, etc.) in newspaper articles for the subsequent processing steps.
- Juxtaposition Analysis: For each entity Lydia identifies what other entities occur near it in an overrepresented way. This analysis is influenced by the intuition that our mental model of an entity largely depends on how the entity relates to other entities. The juxtaposition leads to a significance score for every entity that

co-occurs with another. It can be seen as a time-dependent popularity scoring function, which highlights the current importance in the news.

- Sentiment Analysis: Approximates an article's subjectivity and polarity with a dictionary-based approach.
- Temporal Analysis: Lydia's main database stores all references to entities broken down by article classification. Accordingly, it enables the user to analyze frequency related time-series. Specifically, the user can ask the question, how many occurrences of one entity can be found in each of the categories (news, business, entertainment, sports, other) in a specific time interval.
- Spatial Analysis: Enables the user to see where, in the U.S., people are talking about a specific entity. Lydia shows this information in a heatmap visualization, where the heat is defined as a function of the entity's reference frequency and the sphere of influence of the specific news source. The news source's national influence is determined by its location and publication circulation.

The Lydia project also offers a web front end to access its information: The TextMap website [Tex11b] is an entity search engine providing information about different entities (people, places and things) extracted from the news sources. Figure 2.4 (right side) shows TextMap's reference page of the entity *Barack Obama*. It gives an insight into the entity relation network (left), his reference frequency over time (1^{st} row in the middle), his frequency of appearances in the categories (news, business, entertainment and sports) (2^{nd} row in the middle) and a polarity measure over time (last row in the middle). Additionally, it shows the co-reference scores of the top six co-occurring entities on the left side.

2.3 NewsInsight

Lu et al. present in [LHZ+11] the news exploration system *NewsInsight*. Alike the Lydia approach, NewsInsight extracts various kinds of entities from the news articles, namely persons, organizations, and locations. In the subsequent processing step it clusters the related news with the help of a topic model (Topic Model Clustering will be described in Section 3.1 "Text Data Clustering"). Figure 2.5 shows NewsInsight's architecture.

As Figure 2.5 depicts NewsInsight performs a textual preprocessing (Preprocessing techniques will be described in Section 4.1) and then analyses the data for relational and trend information. Three kinds of relations are examined: between entities, between topics and the relations between topic and entities. Based on the extracted topic model, NewsInsight calculates the similarity between entities or topics as a weight of the relations, and labels those relations with keywords which are highly relevant to both sides

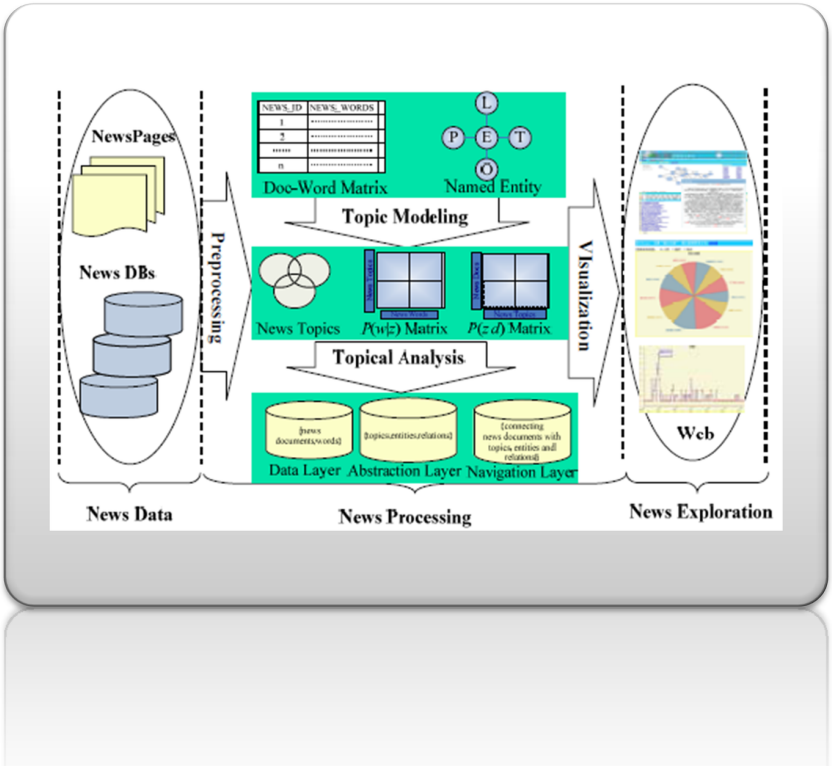


Figure 2.5: NewsInsight Architecture and Processing Pipeline [LHZ+11].

of the relations. A "change trend" is also measured with the number of news related to the entities or topics throughout the timeline.

2.4 NewsBlaster

The Columbia University's NewsBlaster system [MBE+02] is an online news exploration system, which was developed by McKeown et al. in 2001. Since then it provides the user with a topically clustered news summary. The system has six major phases: crawling, article extraction, clustering, summarization, classification, and web page generation. Its main emphasize lies on multilingual multi-document summarization.

Figure 2.6 shows the NewsBlaster online portal. The simple web interface reveals a broad categorization into the six major themes (U.S., World, Finance, Sci/Tech, Entertainment, and Sports). After chosing one article from the overview, the user will be guided to the topic's overview. Here, an enumeration of related articles, top story keywords, crawled images and a story development timeline can be found. The latter visualizes the stream in a list view, containing the article's publication date on the one hand and the article's headline story on the other hand.

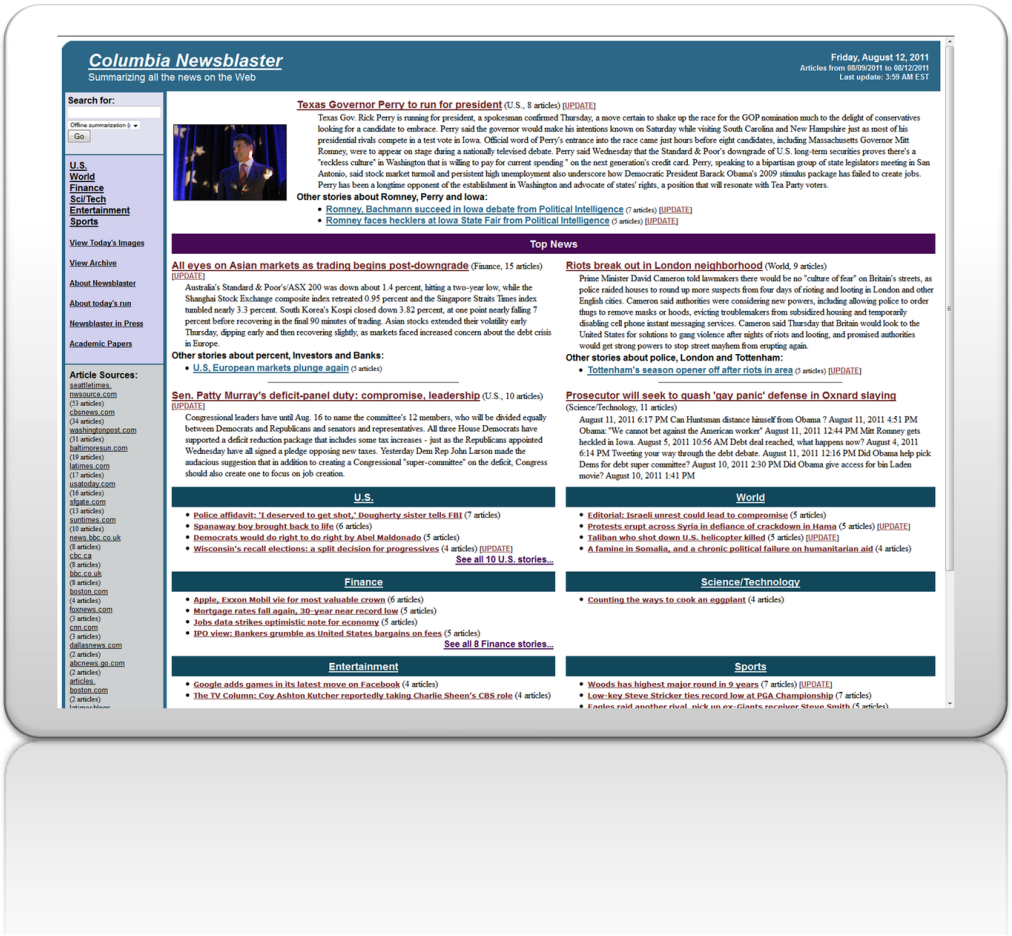


Figure 2.6: NewsBlaster Portal [MBE+02].

From a technical point-of-view the NewsBlaster system appears to be outdated. Yet still, the system is fully functional since 2001 and has thus to be regarded as one of the key enablers for this research field. For its news data clustering the system uses a hierarchical clustering method, which will discussed in Section 3.1.
However, outstanding is NewsBlaster's multi-document summarization scheme, as shown in Figure 2.7. The routing engine selects one of two multi-document summarization systems based on the similarity of the documents in the cluster. Whenever the documents center around one single event happening at one place and roughly at the same time, the Multigen summarization system [MKH+99] is used. Multigen clusters sentences based on similarity (a set of similar sentences defines a theme), and then uses an alignment of parse trees to find the intersection of similar phrases within sentences to form a summary. A more detailed description of the techniques behind Multigen can be found in Section 3.3.

The second summarization system used is DEMS, or Dissimilarity Engine for Multi-document Summarization [SNMS02], which uses a sentence extraction approach to summarization. DEMS can be configured for the biographical task (documents dealing with one event concerning one person) or with a more general event configuration (one event that occurs at different places at different times with different protagonists). DEMS selects sentences that contain interesting or important information, evaluated by a combination of several features that are critical for new-information detection. The evaluation follows traditional heuristics used in single-document summarization. The feature choice will be described in detail in Section 3.3.

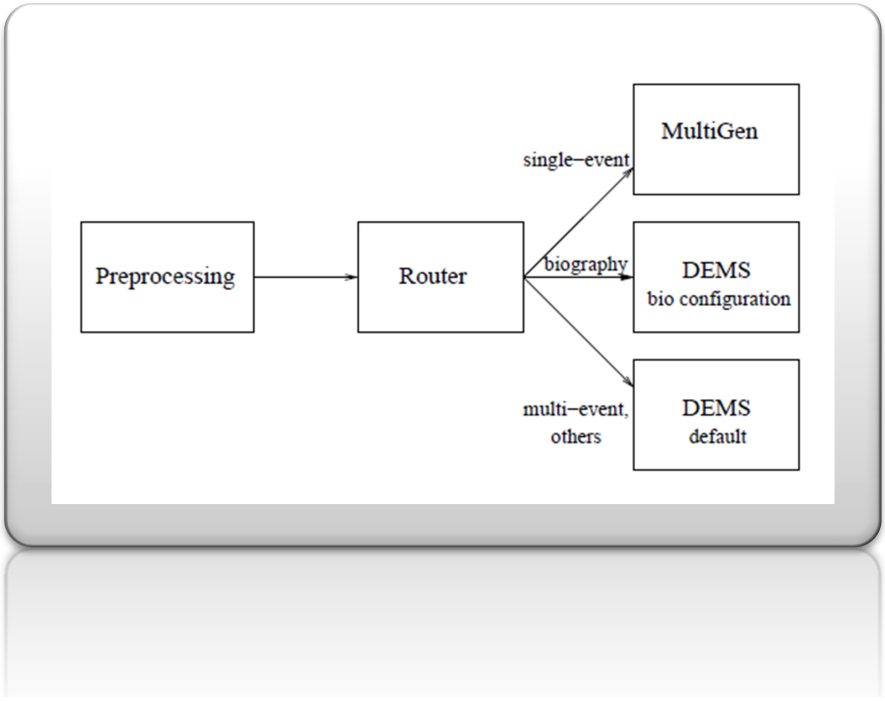


Figure 2.7: NewsBlaster Summarization Scheme [MKH+99].

Chapter 3

News Data Exploration Techniques

News data exploration is a broad subfield of text analysis. Especially here, an intertwined combination of natural language processing-, data mining- and visual analytics approaches can be found. On a very abstract level, the process of news exploration can be visualized in a pipeline as shown in Figure 3.1.

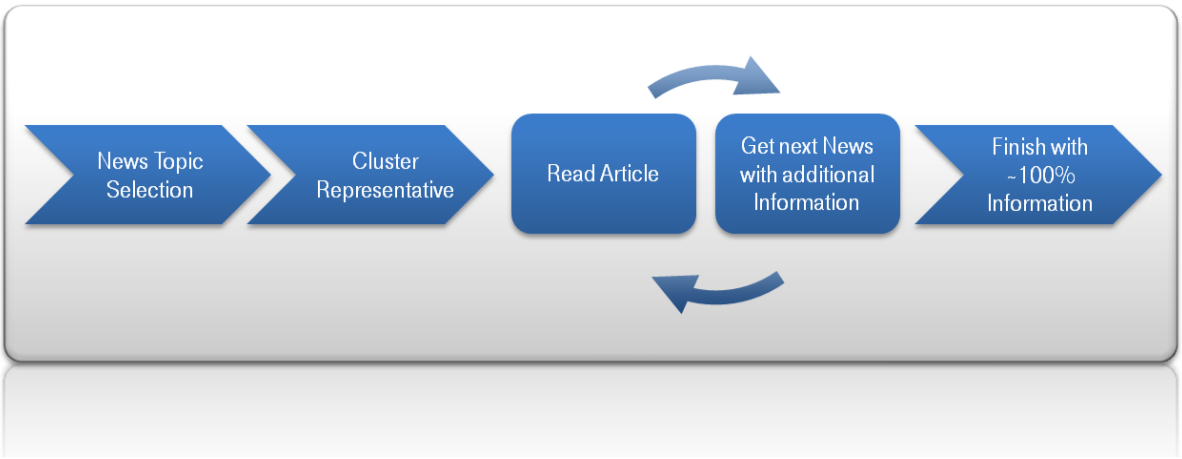


Figure 3.1: News Topic Exploration Pipeline.

This standard pipeline can be found –to some extend– in most news data aggregators. While some of the pipeline sections have sophisticated and technically mature approaches, such as the "News Topic Selection" other sections are underdeveloped and give room to introduce enhanced intelligence into the news reading process (e.g. the "Topic/Cluster Representative"). However, to understand the intent of the pipeline sections and emphasize its extension points, every subsegment must be investigated in detail.

News Topic Selection
The pipeline starts with the "News Topic Selection". Most news aggregators and portals identify news clusters, or topics. These topics are subsequently classified into several categories (e.g. sports, politics, entertainment) and presented in a list view of topics sorted by their publication date. Scrolling through the topic list can be seen as going back in time or approaching the topic in descending importance order, depending on the implementation.
Technically the process of topic identification is challenging. It incorporates keyword/entity extraction (see: Section 3.5) and text data clustering (see: Section 3.1).

Topic/Cluster Representative
While scanning through the overview page of a news portal the users are not only confronted with the news' title. Their decision to read the text is influenced by the corresponding description snippet, too. The description text has two purposes. First, it should give the user basic information about the topic without the need to read full texts. Second, it should function as a decision clue to read the full article. In most news portals the description text extraction mechanisms are rather simple and lead to unsatisfactory results. Their texts are usually a fixed-size excerpt of one news' full-text. Often, it originates from the most recent article of the topic or –which is a better solution– stems from the most relevant article.
This pipeline section gives significantly room for extension. From a user perspective, the best information gain would be achieved if the description text covered the main storyline without distracting with secondary information/aspects. One solution to this problem is to summarize the news article or news topic cluster by automatic means. In this approach the most salient sentences describing the article/text are extracted and used for presentation purposes. Text summarization approaches will be covered in Section 3.3.

Read Article
After choosing one article on the news portal the user is referred to the article's web site. As a result, online news exploration systems lose the data control. They have no chance to track article reading durations, mouse clicks or movements or –in an imaginary optimal case– eye movements. If news exploration systems would be able to incorporate relevance feedback into their news selection/recommendation, they would leverage the user's satisfaction significantly.
As a consequence, a full-fledged news exploration system must incorporate some kind of relevance feedback. Section 3.4 gives an overview about the relevance feedback techniques and how they can be used for the news exploration domain.

Get Next News
In all news exploration systems the user is left with the choice which article to read next

after finishing the first/previous one. This, most noticeably, leads to a redundancy in the reading process, because every news article covers again the same news story. Hence, the user has to skim over the entire new news article to find new aspects/background information. As one better solution for the problem, the reader could be recommended –as an example– to the semantically least similar article or to the article (-part) which covers most of the non-read aspects. News recommendation systems incorporate and combine several news exploration techniques to find the best article. For example, these systems must have some notion of text similarity (see: Section 3.2) so that they can distinguish between similar and dissimilar text. Even more important is that they have to present a ranked result list of articles to the user. This ranking process must be influenced by user feedback so that the system takes the read information into account (see: Section 3.4). Lastly, the system must have means to visually guide the user through the reading process. Some of these visualization techniques, which are interesting for the news domain, are presented in Section 3.6.

Finish Reading
In all news aggregators/portals the users are left with the choice whether or not they shall read more news. This is due to the fact that none of the news exploration systems enable users to keep track of the amount of information they already accessed versus the yet unread information. Tracking this information gain would enable the user to read much more effectively. The Section 3.5 presents some of the approaches in this field and highlights aspect-oriented extraction. This technique enables the user to track his information gain advances.

The following section gives an insight into the techniques behind the news exploration pipeline. It will describe its functionality and relation to the news exploration domain.

3.1 Text Data Clustering

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters) [JMF99]. In the news data domain the definition can transformed to the following equivalent: Groups, or clusters, can be seen as the news' topic, whereas data items are the news articles. The task is then defined as retrieving a number of topic clusters from a news article stream (or its subset), so that each cluster contains only similar articles. All articles, which are not contained in the cluster have to be dissimilar.
News clustering algorithms can be divided into discriminative and generative types. Broadly speaking, discriminative algorithms operate on pairwise similarities between the documents. They optimize a criterion (or objective-) function to produce an optimal

clustering. Generative algorithms, on the other hand, assume an underlying distribution of the data, and maximize the fit of the distribution to produce cluster centroids.

3.1.1 Discriminative Algorithms

A discriminative learning algorithm is a model-based supervised learning algorithm that produces a discriminative model (by directly estimating the conditional probability and the a posteriori probability of the target attribute with respect to the predictor variables) [Min09] [Mel11].

Partitioning Algorithms

The perhaps most common clustering algorithm is spherical K-Means, which is described in most texts about clustering (see for instance [JMF99]). Algorithm 3.1 showcases the basic algorithm.

Algorithm 3.1: K-Means Algorithm

1. Pick k objects at random and let them define k clusters.
2. Calculate cluster representatives.
3. Make new clusters, one per cluster representative.
   Let each text belong to the cluster with the most similar
   cluster representative.
4. Repeat from 2 until a stopping criterion is reached.

The starting point of the K-Means clustering is a random initial partition. This initial partitioning is crucial for the algorithm's final result and influences the performance and result-accuracy likewise. Therefore, many K-Means modifications focus on the initial partitioning step.
In the following step the cluster representative is calculated. In most cases, the mean –or centroid– of the objects is chosen.
The third step recalculates the cluster assignment according to the predefined distance metric/similarity function. For every item the lowest distance to the (updated) cluster representative depicts its cluster assignment. Accordingly, K-Means' standard objective function can be defined as:

$$E = \frac{1}{N} \sum_{x} \left\| \mathbf{x} - \mu_{k(\mathbf{x})} \right\|^2 \tag{3.1}$$

where $\| \cdot \|$ denotes a distance metric (e.g. L_2 norm), $\mathbf{x}$ is a k-dimensional vector representing one data item, $k(\mathbf{x}) = argmin_{k \in \{1,\cdots,K\}} \|\mathbf{x} - \mu_k\|$ is the index of the closest cluster centroid to $\mathbf{x}$, N is the total number of data vectors, and K is the total number of clusters.

As Formula (3.1) depicts, K-Means tries to minimize the mean-squared error, because every article will be assigned to the cluster representative with the lowest distance. For high-dimensional data, such as text documents, the cosine similarity has been shown to be a superior measure to the Euclidean distance. The implication is that the direction of a document vector is more important than its magnitude. Likewise, the objective function has to be changed to:

$$L = \sum_{x} \mathbf{x}^T \mu_{k(\mathbf{x})} \tag{3.2}$$

where $k(\mathbf{x}) = argmax_k \mathbf{x}^T \mu_k$. Here, K-Means tries to maximize the average cosine similarity.
The latter process steps will be repeated as long as the stopping criterion is not reached. The mostly used stopping criterion is the cluster assignment change rate. If it reaches zero, no reassignment is possible and the clusters have stabilized.

Mainly due to its ease of implementation the K-Means algorithm is one of the dominant clustering methods. However, it suffers significantly from a number of drawbacks: First, it relies on a stochastic (random) initialization. Second, it can converge to suboptimal local minimum. Third, it is susceptible to outliers and noise. Fourth, its complexity is $\mathcal{O}(nkl)$, where n is the number of documents in the corpus, k is the desired number of clusters, and l is the number of iterations. And fifth, the numbers of clusters is always fixed to the initially chosen value k.
As already mentioned above several K-Means modifications exist. For example, Zhong et al. [Zho05] suggest an online competitive learning technique, called Online Spherical K-Means Clustering (short *oskmns*). This technique allows to speed up the clustering while achieving similar or better accuracy. More importantly, in an online competitive learning scheme, documents are streamed continuously, which stands in contrast to K-Means' original batch processing behavior. In a batch processing the entire data collection is computed in a single "batch", whereas an online processing allows to add documents $\mathbf{d}$ from a document collection $\mathcal{D}$ to the clustering in an iterative fashion. Every cluster –here called cell– competes for the input and subsequently the "winner" adjusts itself to respond more strongly to future inputs according to a learning rate η. As depicted in Formula (3.2) oskmns tries to minimize the following objective function:

$$J = \sum_{\mathbf{d} \in \mathcal{D}} \mathbf{d}^T \mu_{k(\mathbf{d})} \tag{3.3}$$

where $k(\mathbf{d}) = argmax_k \mathbf{d}^T \mu_k$ is the dot product of the document vector and the centroid k. In the subsequent step, oskmns does not modify the centroid by computing the mean, but incorporates the notion of learning into the process:

$$\mu_{k(\mathbf{d})}^{new} = \frac{\mu_{k(\mathbf{d})}^{old} + \eta\mathbf{d}}{\left\|\mu_{k(\mathbf{d})}^{old} + \eta\mathbf{d}\right\|} \tag{3.4}$$

In this part of the objective function, the cluster representative adapts with a gradually decreasing learning rate η towards the incoming data item $\mathbf{d}$. The learning rate η has the function to control "how much the clusters adjust"[Zho05]. A simulated annealing function has proven to be most successful for η. The function looks as follows:

$$\eta_t = \eta_0 * \left(\frac{\eta_{final}}{\eta_0}\right)^{\frac{t}{M|\mathcal{D}|}} \tag{3.5}$$

where M is desired number of iterations, t is the index of the current iteration, η_{final} is mostly chosen to be 0.01 and η_0 is the initial learning rate factor.
Online spherical K-Means clustering has two primary advantages over spherical K-Means clustering. First, its adaptive online feature makes it to an iteratively adapting clustering function without the need to rerun the algorithm on the entire data set. And second, its convergence is twice as quickly as the normal K-Means clustering algorithm. Despite its near-realtime (depending on the document collection size) features, oskmns cannot achieve the same levels of accuracy as other clustering algorithms.

As a further enhancement of the K-Means algorithm, several modifications try to tackle K-Means' main problem: The optimal number of clusters must be known a priori. A few algorithmic solutions to this problem exist, mainly differing in their approach. While some algorithms, such as *Random Partition* and *Forgy* (both described in [HE02]) try to find the right number of clusters by a data set sampling other approaches, such as *SPARCL* [VHSZ09] or *Split and Merge K-Means* [MG09], start with a high number of cluster representatives and merge them iteratively.
Especially, Chaoji's approach *SPARCL*, or Shape Based Clustering (presented in [VHSZ09]) should be emphasized. It is a simple and scalable algorithm for finding clusters with arbitrary shapes and sizes. Its outstanding advantage is that it has linear space and time complexity. The algorithm mainly consists of two stages: The first stage runs a carefully initialized version of the K-Means algorithm to generate many small seed clusters. The second stage iteratively merges the generated clusters to obtain the final arbitrary shaped clusters.

Hierarchical Clustering Algorithms
In the most common sense, hierarchical or agglomerative clustering algorithms begin their clustering process by producing many clusters of smaller sizes and then iteratively merge them to produce bigger clusters. As a result of the algorithm, a density histogram (dendogram) representation of the underlying data set is built in a greedy manner. These

algorithms need have a notion of cluster similarity to combine the clusters. The hierarchical aspect of these clusters is that higher-level clusters (more abstract) are composed of lower-level clusters (more specific). This may either be a single document, a part of the document corpus or –as the final product– the entire document corpus itself. The algorithm is depicted in Algorithm 3.2.

Algorithm 3.2: Hierarchical Clustering Algorithm

1. Construct one cluster for each document.
2. Join the t most similar clusters.
3. Repeat 2 until a stopping criterion is reached.

Hierarchical clustering algorithms differ in their way of characterizing the similarity between clusters, as Figure 3.2 showcases:

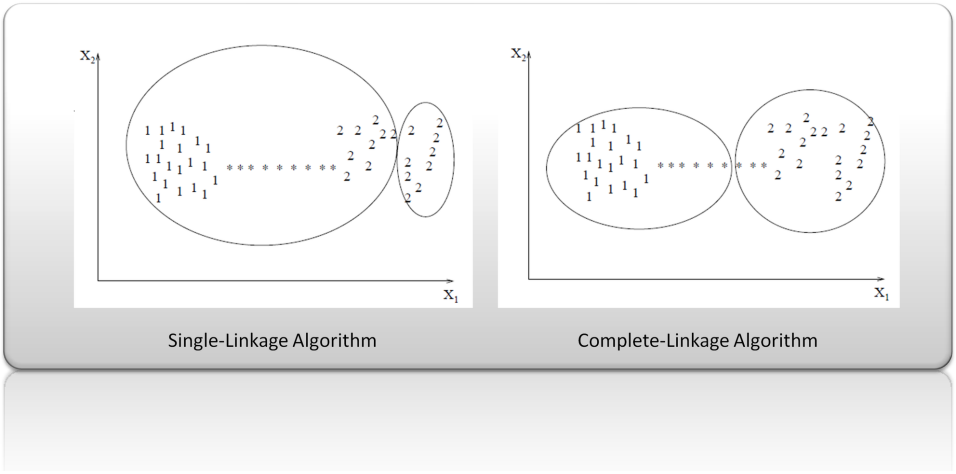


Figure 3.2: Single-Linkage versus Complete-Linkage Results.

Single-Linkage algorithms define the distance between two clusters as the minimum of the pairwise distances. Equation (3.6) depicts the optimization formula:

$$\min \{ d(a, b) : a \in A,\, b \in B \} \tag{3.6}$$

Complete-Linkage algorithms, on the other hand, define the distance between two clusters as the maximum of the distances between all pairs of patterns drawn from the two clusters. Equation (3.7) depicts the related optimization formula:

$$\max \{ d(a, b) : a \in A,\, b \in B \} \tag{3.7}$$

In addition to single-linkage and complete-linkage, other linkage criteria exist as well. For example, *Average-Linkage* calculates the mean distance for each pattern to all other patterns, *Centroid-Linkage* computes the distances of every pattern to all computed cluster centroids, *Intra-cluster variance* takes the sum of all cluster variances into account, or *Ward's criterion* measures the increase of variance for the clusters being merged.

The final results of the mentioned linkage-based clustering algorithms tend to be very different. For instance, the complete-linkage algorithm produces mostly very compact clusters, whereas the single-linkage algorithm, by contrast, has the tendency to produce clusters that are elongated (this chaining effect is described in [Nag68]). As a general comment, one can say that single-linkage algorithms are more versatile than complete-linkage algorithms, since they are able to extract the clusters, which share the same centroid, as shown in Figure 3.2. On the other side, one can also see that complete-linkage algorithms tend to produce more useful dendograms (or hierarchies) than the single-linkage algorithm.
All in all, the time complexity of all agglomerative hierarchical clustering algorithms is $\mathcal{O}(n^2)$, since they have to compute the pairwise similarity between all objects to find the most similar item.

3.1.2 Generative Algorithms

A generative learning algorithm is a model-based learning algorithm that directly estimates the prior probability of the target class and predictor variables [Mel11].

Probabilistic Topic Models
Probabilistic topic models aim is to discover the hidden thematic structure in a document collection. The basic idea behind a topic model is that every document is composed from a mixture of topics. Each topic has a probability distribution over all known words. A generative model specifies to which extend the topics are contained within the document. Thus, it works as a probabilistic function by which documents can be created. In accordance to this statement, a new document can be created by simply choosing random distribution of topics and use the words from the selected topic. Figure 3.3 illustrates the concept in a geometric interpretation. For a clustering task, this process can be inverted. Then, the set of topics defining all articles has to be inferred from the document corpus.
Several algorithms exist in the field of text processing. An exemplary excerpt will be given in the following.

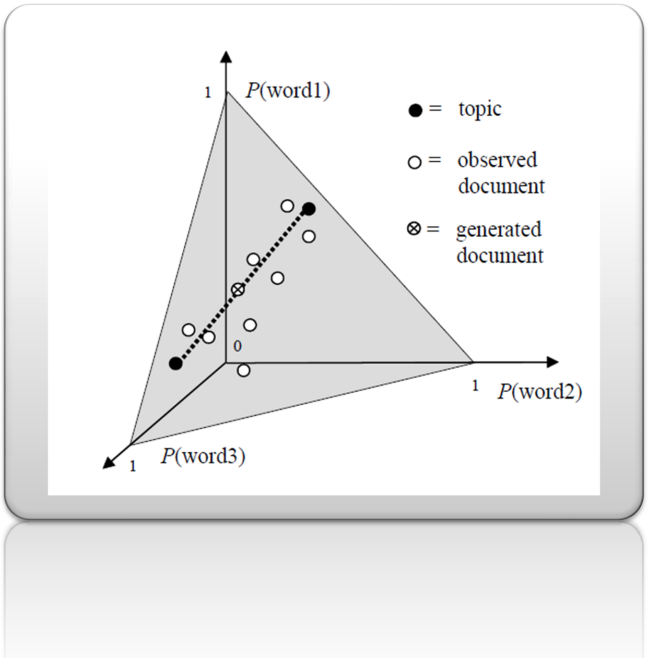


Figure 3.3: Geometric Interpretation of Topic Models [SG06].

Latent Dirichlet Allocation Approach

Latent Dirichlet Allocation, short LDA, is a statistical model that tries to capture the topic proportions and word-topic assignments within a document. The basic assumption for LDA is that documents contains a number of topics, which are –as noted above– a distribution over words, and the number of topics is fixed and known a priori within the document selection/corpus.

LDA can be described by an imaginary random generative process by which the document arose:

Algorithm 3.3: Generative Algorithm LDA

```
1. Randomly choose a distribution over topics.
2. For each word in the document
  (a) Randomly choose a topic from the distribution
    over topics in step #1.
  (b) Randomly choose a word from the corresponding
    distribution over the topic's vocabulary.
```

This statistical model reflects the basic idea of LDA, since it represents multiple topics with a different proportional importance within the text. Each document contains a number of topics (reflected by Step 1) and each word in the document is

drawn from one of the topics (see: Step 2b), where the selected topic is chosen from the per-document distribution over topics (see: Step 2a). Adapted from [Ble11].

The description above treats the document text as a randomly generated distribution. However, probabilistic modeling approaches, such as LDA, treat the data as a generated result controlled by *observed* and *hidden variables*. The data analysis task is therefore to find a conditional distribution of the hidden variables given the observed variables.
According to Blei's work [Ble11], we define LDA more formally with the following notation. The topics are denoted by $\beta_{1:K}$, where each β_k is a distribution over the vocabulary. The topic proportions for the d^{th} document are θ_d, where $\theta_{d,k}$ is the topic proportion for topic k in document d. The topic assignments for the d^{th} document are z_d, where $z_{d,n}$ is the topic assignment for the n^{th} word in document d. Finally, the observed words for document d are w_d, where $w_{d,n}$ is the n^{th} word in document d, which is an element from the entire corpus vocabulary. With this notation, the generative process for LDA corresponds to the following joint distribution of the hidden and observed variables, also called posterior distribution:

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \left(\prod_{n=1}^{N} p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid \beta_{1:K}, z_{d,n}) \right) \tag{3.8}$$

This distribution reveals a number of dependencies. For example, the topic assignment function $z_{d,n}$ depends on the per-document topic distribution θ_d, or the observed word $w_{d,n}$ depends on the topic assignment $z_{d,n}$ and all of the topics $\beta_{1:K}$[Ble11]. Theses dependencies have a general impact on the document-to-topic assignment task as it is required for news data clustering. More specifically, the number of topics has to be known a priori, as well as the assignment of words to these specific topics. Yet still, the largest drawback of LDA is its computational complexity. Theoretically the distribution can only be computed by summing the joint distribution over every possible instantiation of the hidden topic structure. Since the number of possibilities is exponentially large, the computation is not accomplishable for every topic structure. In order to bring the Formula (3.8) to a computational acceptable runtime, an approximate distribution over the latent topic structure has to be found. Generally two major approaches exist for this topic modeling problem:

- Sampling-based Approaches: Sampling-based approaches estimate the parameters of the LDA model given the two Dirichlet priors and a fixed number of topics. The most common approach is *Gibbs sampling*. It constructs a Markov chain of

random variables (each dependent on the previous) and limits the distribution parameter space accordingly. The iterative Gibbs sampling method simulates a high-dimensional distribution by sampling a lower-dimensional subset of variables, where each subset is conditionally dependent on the latter ones. This process proceeds until the sampled values approximate the target distribution. More information on Gibbs sampling is given in [GS04], [GGRS96], and [BLP+04]

- Variational Approaches: Variational algorithms are a deterministic alternative to the latter mentioned Gibbs sampling. Here, rather than estimating the posterior distribution with samples, the method uses a parameterized family of distributions over the hidden structure and finds a member of that family that is closest to the posterior. Hence, the problem is transformed into an optimization task that approximates the posterior distribution appropriately. More information on this LDA approximation technique can be found in [BNJ03]. Hoffman et al. present in [HBB10] a significantly faster alternative to Gibbs sampling. Their work presents an *Online Variational Bayes* algorithm for LDA. This online stochastic optimization introduces a natural gradient step, which can be shown to converge to a local optimum of the variational Bayes objective function. The algorithm is suitable for analyzing large amount of document collections, as well as those arriving in a stream fashion.

3.2 Textual Similarity Measures

Text clustering methods, such as the ones outlined in Section 3.1, most text summarization methods (see: Section 3.3), statistical translation models, or raw text comparison mechanisms are only a part of the application areas that inevitably require the notion of similarity/distance between two text sequences. Judging the similarity between natural language sentences is a critical performance influencer for most approaches that can lead a good-working method to horrible results.
In the most general definition, an effective similarity measure should determine whether two sentences are equivalent or not. Nonetheless, this definition does not take into account on which semantical level a comparison takes place. Figure 3.4 depicts the subcategorization into three general clusters, which will also function as a structuring for the following chapter.
First, the word-level similarity measures compute a similarity score based on the number of shared words within the two sentences. Second, structure-level similarity measures utilize or enhance the latter techniques with linguistic features, such as the syntactic composition. The third group of textual similarity measures incorporates semantics and background knowledge into the computation process.

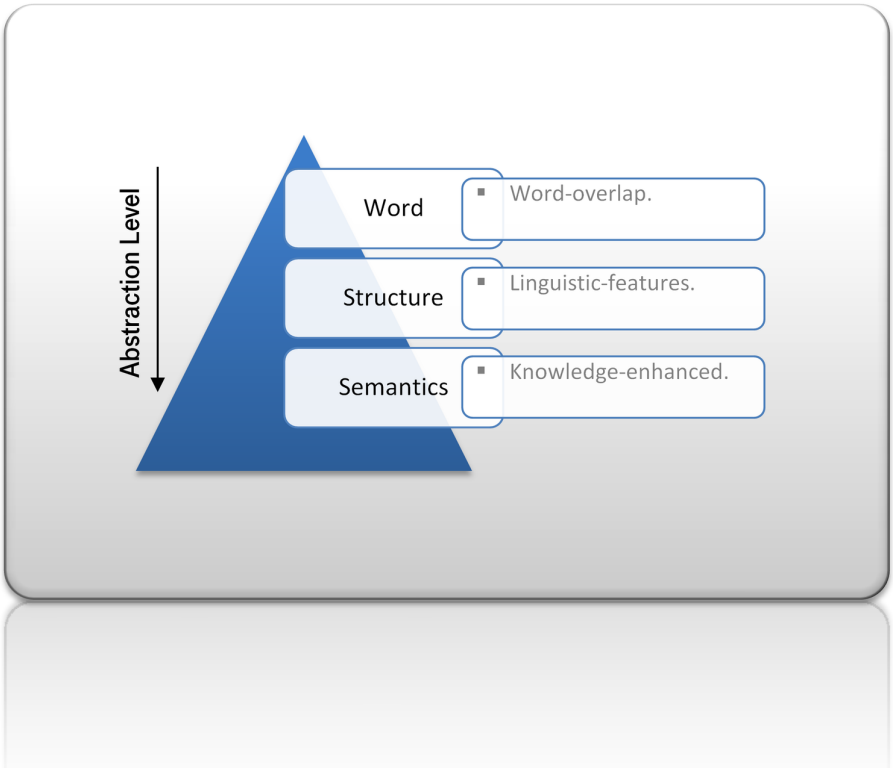


Figure 3.4: Levels of Analysis for Text Similarity.

3.2.1 Word Overlap Measures

Word overlap measures are a family of combinatorial similarity measures that compute the similarity of two sentences based on their shared words. Mainly due to their implementation simplicity one can find these algorithms in a wide range of systems.
In the most cases, word-overlap measures are basing their work on a vector representation of the document (Bag-of-Words model). We denote by $w(D)$ the vector derived from document D, with one component in the vector for each dictionary term.

Jaccard Similarity Coefficient
The Jaccard similarity coefficient is a measure of the word overlap between two sentences or documents. The measure is defined as follows:

$$S_1(D_x, D_y) = \frac{|w(D_x) \cap w(D_y)|}{|w(D_x) \cup w(D_y)|} = \frac{tf_{xy}}{tf_x + tf_y - tf_{xy}} \tag{3.9}$$

Here, $w(D_x)$ and $w(D_y)$ are the set of document-words on which the similarity is calculated. Hence, the Equation (3.9) defines the similarity as the cardinality of the intersection of words –or common words– divided by the cardinality of the union of words. A further possibility to calculate the Jaccard similarity, or also called Jaccard index, is given, too. In this notion tf_x and tf_y defines the term-frequency of the words in $w(D_x)$, respectively $w(D_y)$, and tf_{xy} is the collocation frequency of both terms.
A value of 0 will be produced by the Jaccard similarity coefficient if no word-overlap is recognized. A value of 1 represents a perfect agreement. Higher numbers indicate a better agreement/higher similarity.

Dice Similarity
Like the Jaccard similarity measure, the Dice coefficient measures set agreement. The formula is given in Equation (3.10):

$$S_2(D_x, D_y) = \frac{2\,|w(D_x) \cap w(D_y)|}{|w(D_x)| + |w(D_y)|} = \frac{2tf_{xy}}{tf_x + tf_y} \tag{3.10}$$

where $w(D_x)$ and $w(D_y)$ are the two word sets. More simply, this formula represents the size of the union of 2 sets divided by the average size of the two sets. Alike Jaccard, a value of 0 indicates no overlap; a value of 1 indicates a perfect agreement.

Cosine Similarity
The cosine similarity has a different motivation in comparison to the Jaccard- or Dice similarity. In the case of the cosine similarity, the angular difference between the both documents is regarded as a similarity measure, rather than the magnitude of the resulting vector. Equation (3.11) depicts the cosine similarity formula:

$$S_3(D_x, D_y) = \frac{2\,|w(D_x) \cap w(D_y)|}{\sqrt{|w(D_x)|} * \sqrt{|w(D_y)|}} = \frac{tf_{xy}}{\sqrt{tf_x * tf_y}} \tag{3.11}$$

Representing similarity as a measure of angular difference is valid, because the sparse high-dimensional vector-representation of a document can be regarded to be embedded into the high-dimensional vector space with the origin $[0, \ldots, 0]$. For text documents this has one clear advantage. In the theoretical case, in which one document would compared to the concatenation of another document with itself, the cosine calculation would result in the same value as without the concatenation ($S_3(D_1, (D_2 \circ D_2)) = S_3(D_1, D_2)$). Hence, not the amount of similar words is of interest, it is more the direction of each document's vector representation in the high-dimensional space.

Average Conditional Probability

The Average Conditional Probability (shown in Equation (3.12)) depicts the average value of two conditional probabilities $P(x \mid y)$ and $P(y \mid x)$.

$$S_4(x, y) = \frac{P(x \mid y)P(y \mid x)}{2} = \frac{1}{2}\left(\frac{P(x,y)}{P(x)} + \frac{P(x,y)}{P(y)}\right) = \frac{tf_{xy}}{2}\left(\frac{1}{tf_x} + \frac{1}{tf_y}\right) \tag{3.12}$$

where, x is one a word in the document D_x, y is one a word in D_y, $P(x) = tf_x/N$, $P(y) = tf_y/N$, $P(x, y) = tf_{xy}/N$.
$P(x \mid y)$ reveals how much the term y depends on term x. More specifically, the more frequently term y co-occurs with the term x, the higher is the dependence of y on x regardless of x's frequency. Analog to the latter statement, $P(y \mid x)$ denotes the dependence of x on y and thus the opposite case of $P(x \mid y)$.
Conditional probabilities follow the intuition that collocation has a different importance according to each term's frequency size [Sin91]. If the term x occurs much more frequently than term y the collocation is more important for term y than for term x. Resulting from this idea, Park and Choi present in [PC96] the Average Conditional Probability in order to normalize the conditional probabilities of two terms. Note that the Formula in Equation (3.12) depicts a per-word similarity measure. It can be enhanced to a sentence-similarity measure by a simple aggregation function over all terms.

Normalized Mutual Information

The Normalized Mutual Information, short NMI, is defined by dividing the mutual information of term x and y with the maximum mutual information value [KKC94]. The formula uses the notion of mutual information, which depicts the strength of the statistical dependency between the terms. Equation (3.13) depicts the NMI formula:

$$S_5(x,y) = \frac{MI(x,y)}{\max MI} = \log\left(\frac{P(x,y)}{P(x)P(y)}\right) / \log N = \log\left(\frac{N * tf_{xy}}{tf_x * tf_y}\right) / \log N \tag{3.13}$$

where $\max MI = \log N$.
If the mutual information of any two terms is negative, two terms are weakly relevant. Thus, the highest possible measure for NMI is 1 depicting the highest information gain.

IDF Overlap Similarity
Metzler et al. define in [MBC+05] a baseline measure. This simplistic word overlap fraction calculates the proportion of words in D_x also appearing in sentence or document D_y and normalizes the overlap by the sentence length $|D_x|$. This baseline formula is the first part of Equation (3.14). Additionally, Metzler enhances the baseline formula with a score, taking the inverse document frequency (Idf) into account. The Idf score of a word depicts its relative importance within the document selection or can be seen as a measure of discriminability. A word with a low document frequency occurs in few documents and is hence more appropriate to use than a word that is contained in nearly every document. Equation (3.14) depicts Metzler's Idf overlap similarity:

$$S_6(D_x, D_y) = \frac{|w(D_x) \cap w(D_y)|}{|D_x|} \left(\sum_{w \in (w(D_x) \cup w(D_y))} \log \frac{N}{df_w} \right) \tag{3.14}$$

where df_w depicts the amount of documents in which w occurs.
This model follows the intuition that high Idf terms are typically strong indicators of shared heritage between two sentences in comparison to low Idf terms [MBC+05].

TF-IDF Vector Similarity
Tf-Idf measures are a broad class of similarity functions. Typically, they are used to approximate/measure the relevance and similarity between two sentences. The fundamental idea is that the more frequently a word appears, the more representative is this word for the topic (known as the *term-frequency*). In addition, the less frequently a term appears in the entire document collection, the greater its power to discriminate between interesting and uninteresting sentences (known as the *inverse document-frequency*). The standard Tf-Idf vector similarity makes use of the cosine similarity, as shown in Section 3.2.1 between vector representation of two sentences.

Allan et al. present in [AWB03] a measure for finding topically similar sentences, which consistently –but not significantly– outperforms language modeling based approaches. The similarity function is:

$$S_7(D_x, D_y) = \left(\sum_{w \in (w(D_x) \cap w(D_y))} \log(tf_{w,D_x} + 1) \log(tf_{w,D_y} + 1) \log \frac{N+1}{df_w + 0.5} \right) \quad (3.15)$$

where tf_{w,D_x} is the number of occurrences of w in D_x; tf_{w,D_y} is the number of times term w appears in D_y, N is the total number of documents in the collection, and df_w is the document-frequency of the word w.

Further variations of Tf-Idf similarity measures are given in [HZZ03]. Hoad et al. propose a measure for identifying plagiarized documents or co-derivation. It has been shown to perform effectively for such applications. The so called *identity score* is built from the sum of inverse document frequencies of the words, which appear in both sentences. This score is normalized by the overall lengths of the sentences and the relative frequency of a word between the two sentences:

$$S_8(D_x, D_y) = \frac{1}{1 + \frac{\max(|D_x|,|D_y|)}{\min(|D_x|,|D_y|)}} \sum_{w \in (w(D_x) \cap w(D_y))} \frac{N/df_w}{1 + |tf_{w,D_x} - tf_{w,D_y}|} \quad (3.16)$$

Phrasal Overlap Measure

Phrasal overlap measures, such as the one described by Banerjee and Pedersen in [BP03], are motivated by the fact that the classical *Bag-of-Words* model does not take the differences between a single word and multi-word phrases into account. As a result of this, finding the overlap of n-word phrases will be much rarer to retrieve than one word overlap. Consequently, the combination of words, also called n-grams, will have a more discriminative power than simple one-gram overlap measures. Banerjee and Pedersen suggest a phrasal-overlap measure as depicted in Equation (3.17):

$$S_9(D_x, D_y) = \sum_{i=1}^{n} \sum_{m} i^2 \quad (3.17)$$

where m is a number of i-word phrases that appear in the sentence pairs. Equation (3.17) can be normalized by first dividing by the sum of sentences length and apply the hyperbolic tangent function. The hyperbolic tangent function proves to minimize the effect of outliers in the result. Equation (3.18) displays the normalized function developed by Ponzetto and Strubbe [PS07]:

$$S_{10}(D_x, D_y) = \tanh \left(\frac{S_9(D_x, D_y)}{|D_x| + |D_y|} \right) \quad (3.18)$$

3.2.2 Structural and Linguistic Measures

In addition to the "raw" word overlap measures, showcased in Section 3.2.1, the utilization of linguistic knowledge can prove to be useful to determine sentence similarity scores.

Word Order Similarity

A sentence's semantic intention is not only depicted by its wording. Rather so, its word composition incorporates primary syntactic information, which is inevitable to understand the semantics, as well. As one example, *"A quick brown dog jumps over the lazy fox."* describes a semantically different aspect as *"A quick brown fox jumps over the lazy dog."* Word similarity measures will compute a very high similarity (specifically, a score of 1), since the choice of words is equivalent and the Bag-of-Words model does not include a word-ordering per se. Word order similarity measures however, such as the one described by Li et al. in [LMB+06], define the word composition as a normalized difference of two sentence's word order:

$$S_{11}(D_x, D_y) = 1 - \frac{\|r_x - r_y\|}{\|r_x + r_y\|} \tag{3.19}$$

where r_x and r_y are the word order vectors of the sentences/documents D_x, respectively D_y. The word order vector is constructed as follows: A unique index is assigned to every word in the joint word set of D_x and D_y. Then for each word in both sentences, the unique index represents the word in the vector. In the example from above the two vectors will be:

$$\begin{aligned} D_x &= \text{"A quick brown dog jumps over the lazy fox."} \\ r_x &= 1, 2, 3, 4, 5, 6, 7, 8, 9 \\ D_y &= \text{"A quick brown fox jumps over the lazy dog."} \\ r_y &= 1, 2, 3, 9, 5, 6, 7, 8, 4 \end{aligned} \tag{3.20}$$

A different way to calculate word order similarity is presented by Zhang et al. in [ZSWH11]. Their word order similarity takes not only the difference of word orders from one word into account, but rather compares sequences of word orders to each other. Zhang et al. suggest to represent the sentences as a *sequential network* that shows the *sequential* relations. Thus, a sentence will be represented by the vector:

$$L(D_x) = (w_{x1}, w_{x2}), (w_{x1}, w_{x3}), \ldots, (w_{x(i-1)}, w_{xi}) \tag{3.21}$$

where $(w_x, w_y) \in L(D_x) \cup L(D_y)$. The similarity between D_x and D_y can then be

calculated based on the orders of words by the following Formula (3.22):

$$S_{12}(D_x, D_y) = \frac{|L(D_x) \cap L(D_y)|}{|L(D_x) \cup L(D_y)|} \tag{3.22}$$

Word Distance Similarity
Considering the distance between word pairs is another statistical mean for assessing structural characteristics of sentences. For example, Zhang et al. present an enhanced version of their *word order similarity* in [ZSWH11]. The sentence similarity measure operates on a distance network of words within the sentence:

$$L(D_x) = (w_{x1}, w_{x2}, d(w_{x1}, w_{x2})), (w_{x1}, w_{x3}, d(w_{x1}, w_{x3})), \ldots, (w_{x(i-1)}, w_{xi}, d(w_{x(i-1)}, w_{xi})) \tag{3.23}$$

where $d(w_x, w_y)$ denotes the word order distance of word w_x to word w_y. The similarity between two sentences D_x and D_y can then be calculated based by the following Formula (3.24):

$$S_{13}(D_x, D_y) = \frac{\sum_{w_{i,j} \in W(D_x, D_y)} d(x_i, x_j) d(y_i, y_j)}{\sqrt{\sum_{w_{x_i, x_j}} d^2(D_x)} \sqrt{\sum_{w_{y_i, y_j}} d^2(D_y)}} \tag{3.24}$$

where $W(D_x, D_y) = \{(w_{x_a, x_b}) | 1 \leq a \leq b \leq i\} \cap \{(w_{y_p, y_q}) | 1 \leq p \leq q \leq j\}$ and $1 \leq i \leq j \leq |W(D_x, D_y)|$.

3.2.3 Semantic Relation Measures

Enhancing the textual similarity calculation with background knowledge improves the retrieval of topically related documents significantly. The strongest level of textual information is incorporated on a semantic level, where each word's meaning contributes to the overall information. Semantic relatedness between two lexically expressed concepts should therefore have a high score if the concepts have a close relation in a comprehensive lexical knowledge base resource.

Semantic Knowledge Base Resources

The primary knowledge repositories for semantical word and concept coherences are WordNet [MBF+90], Wikipedia (e.g. used in [PS07]), or web search results (as used in [SH06]). These resources vary significantly in several aspects, which are defined by the wrap-up communique of the Ontology Summit 2007 (adapted from [BO07]):

1. Expressiveness: Expressiveness is a property of the knowledge representation language (KRL) which describes the extent and ease with which the KRL can describe increasingly complex semantics, cf. propositional logic, description logic(s), first order logic, sorted logics, modal logics, etc.

2. Structure: Structure is a property of the ontology, which records how elaborate (or well organized) the semantics encoded by the ontology are. Viewed from a graph theoretic perspective, the level of structure might be either a simple set of terms (glossary), a tree structures (taxonomy), a directed acyclic graph, e.g., a partial order (faceted classification schemes), or an arbitrary directed graph (e.g., RDF).

3. Granularity: The granularity dimension concerns the level of detail at which the ontology is specified. A crude measure of granularity would be the number of concepts (nodes) and the number relation instances (links or edges in a graph representation). However, this fails to recognize that some ontologies may have larger scopes or domains than others.

4. Intent: Intended use is the dimension which records the original purpose(s) of the ontology. These may include semantically informed search, data semantics specification for databases or data entries, data integration across multiple data sources, agent communication languages, controlled vocabularies for recording medical diagnoses, etc.

5. Automated Reasoning: Automated reasoning is a dimension which records the extent to which it is anticipated that an ontology will be used by automated reasoning software, e.g., for question answering, etc. If so, then one would expect that the ontology would likely be encoded as using some form of logic, e.g., first order logic.

6. Prescriptive vs. Descriptive: Prescriptive vs. Descriptive is a dimension which characterizes whether the intent of the ontology developer is simply to describe contemporary semantic usage without much regard as to the scientific correctness of the encoded knowledge (e.g., a whale might, in common parlance, be described as a large fish.) Examples of such descriptive ontologies include folksonomies and most linguistic ontologies. Alternatively, an ontology may be intended as a normative prescriptive document whose correctness is considerable concern, e.g., a whale is a mammal not a fish.

In the following, the most relevant lexical knowledge base resources for semantic relation measures will be presented in descending order.

The Princeton University project *WordNet* is an online lexical reference system

inspired by the psycho-linguistic theories of the human lexical memory. Nouns, verbs, adjectives and adverbs are organized into synonym sets, short *synsets*. Each synset represents a closely related lexical concept.
Noun synsets are related to each other through *hypernymy* (generalization), *hyponymy* (specialization), *holonymy* (whole of) and *meronymy* (part of) relations. Of these, (hypernymy, hyponymy) and (meronymy, holonymy) are seen as complementary pairs. An example of the structure/taxonomy is depicted in Figure 3.5.

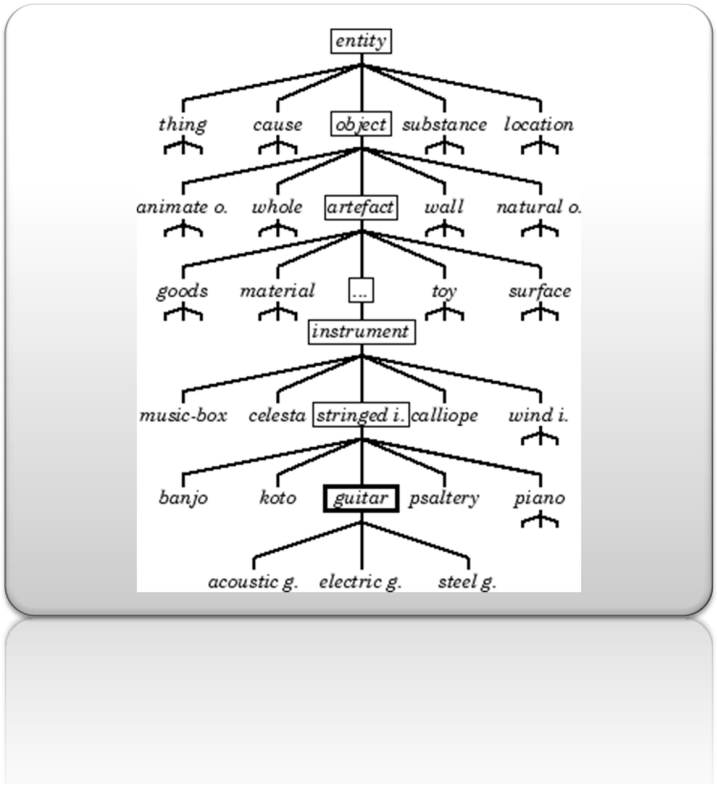


Figure 3.5: WordNet's primary Taxonomy [Wor11].

The verb and adjective synsets are very sparsely connected with each other. No relation is available between noun and verb synsets. However, 4.500 adjective synsets are related to noun synsets with *pertainyms* (pertaining to) and *attra* (attributed with) relations.
In the current version 3.1 the knowledge base contains an overall of 117.000 synsets.

Another primary source of semantic relation knowledge is *Wikipedia*. Wikipedia is a multi-lingual web based encyclopedia, collaboratively generated, edited and revised by volunteers. Wikipedia provides a large domain-independent encyclopedic repository. Every article in the Wikipedia knowledge base is categorized into a predefined set of

categories. Since May 2004 these categories have been enhanced to a semantic network, which is showcased in Figure 3.6.

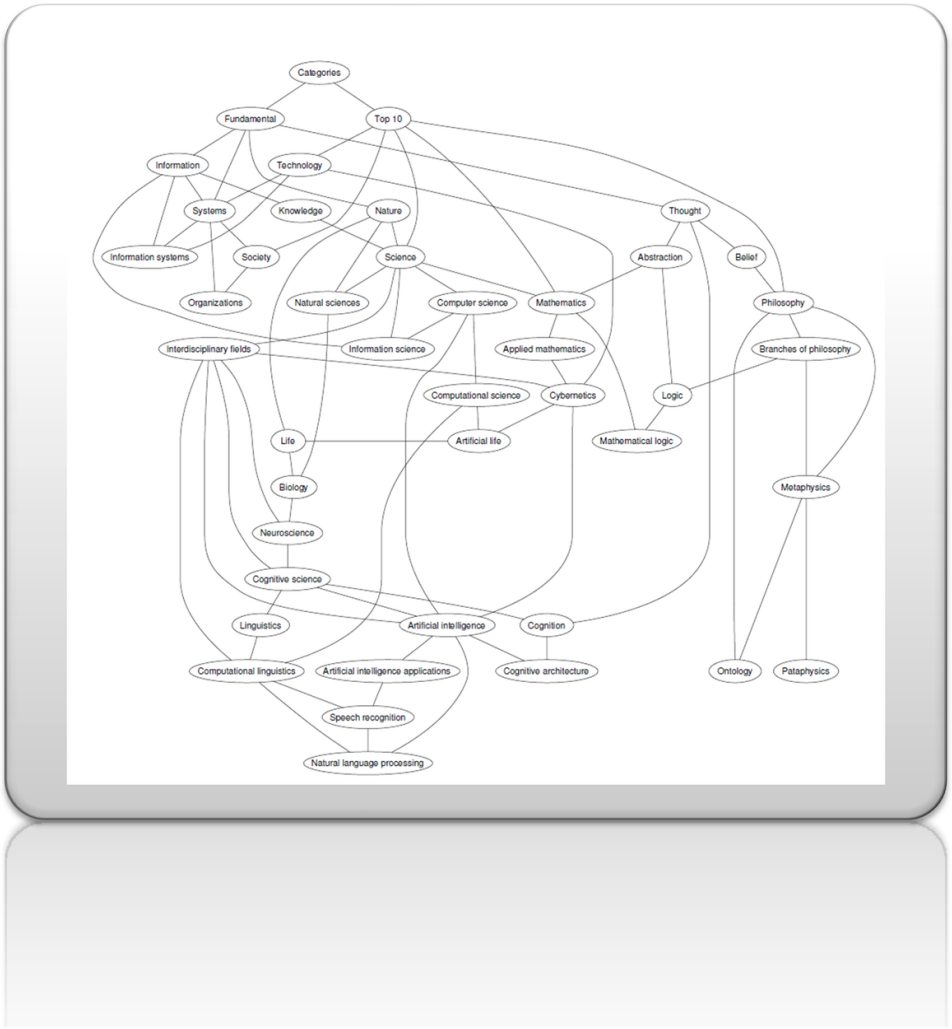


Figure 3.6: Wikipedia's Categories form a Semantic Network [PS07].

In addition to the categorization into the main taxonomy, Wikipedia's article pages reveal various relations between each other (adapted from [PS07]):

1. Redirect Pages: These pages are used to redirect the query to the actual article page containing information about the entity denoted by the query. This is used to point alternative expressions for an entity to the same article, and accordingly models synonymy. Examples include *car* redirecting to the *automobile* and *sickness* referring to *disease* pages.

2. Disambiguation Pages: These pages collect links for a number of possible entities the original query could be pointed to. This models homonymy. For instance, the page *bush* contains links to the pages *shrub*, *bush louisiana*, *George H.W. Bush* and *George W. Bush*.

3. Internal Links: Articles mentioning other encyclopedic entries point to them through internal hyperlinks. This models article cross-references. For instance, the page *pataphysics* contains links to the term inventor, *Alfred Jarry*, followers such as *Raymond Queneau*, as well as distinctive elements of the philosophy such as *nonsensical* and *language*.

The 2001 founded non-profit project Wikipedia contains approximately 19 million articles in 280 languages (Status September 2011).

The last highly important source of semantic relation knowledge is the largest corpus itself, the World-Wide-Web. While taxonomy-enhanced knowledge sources have to be designed carefully and the entering of high quality contents in these structures by knowledgeable human experts comes at great cost, the overall information entered is minute compared to what is available on the WWW.
The rise of the WWW has enticed millions of users to type in trillions of characters to create billions of web pages of on average low quality contents. The sheer mass of the information about almost every conceivable topic makes it likely that the majority or average is meaningful in a low-quality approximate sense [CV07].

WordNet-based measures

WordNet-based measures can be subsumed under the following two categories: First, information-based approaches, such as in Resnik [Res95], Lin [Lin98] or Jiang and Conrath [JC97], which rely on information content calculations. And second, path-length based approaches, such as Leacock and Chodorow [LC98] or Wu and Palmer [WP94], calculating the (relative) path length between two concepts/words. Note that most of the presented techniques measure the semantic relation between concepts, which is the inevitable basis for the sentence similarity scoring.

Resnik Information-Content Measure

Resnik defined in 1995 an information content based similarity measure. Information content is a quantification of the informativeness of a concept. It is defined as in Equation (3.25):

$$IC(c) = -\log\left(\frac{freq(c)}{N}\right) \tag{3.25}$$

where c denotes a concept, $freq(c)/N = P(c)$ or the probability of encountering an instance of the concept c in the corpus of size N. Intuitively, the probabilities of a concept are monotonically increasing as one moves up the taxonomy. Specifically, if c_1 is in a hyponym-relation to c_2 (c_1 IS-A c_2) this will be reflected by $P(c_1) \leq P(c_2)$. Figure 3.7 showcases one example: $P(nickel) \leq P(coin)$. Thus, it is intuitively logic that the information content of a concept c decreases as its probability increases. In other words the more abstract a concept, the lower is its informativeness. Using this logic, Resnik presented the following similarity Formula (3.26):

$$Sim_1(c_1, c_2) = \max_{c \in LCS(c_1,c_2)} (IC(c)) \tag{3.26}$$

In the Equation (3.26) Resnik makes use of the *least common subsumer* (LCS), which is defined as the most specific concept that is ancestor of both c_1 and c_2. As a consequence, in taxonomies with a unique top node, as in WordNet's structure, the probability of the root will be $P(root) = 1$. Consequently, its information content is $-\log(P(root)) = -\log(P(1)) = 0$.
In practice, calculating the concept similarity will be unlikely in comparison to calculating the semantic word similarity. Therefore, one can define $s(w)$ as the set of concepts in the taxonomy, which are senses of the word w and the Resnik semantic word similarity is therefore:

$$S_{14}(D_{x1}, D_{x2}) = \max_{c1 \in s(x1), c2 \in s(x2)} (Sim_1(c1, c2)) \tag{3.27}$$

where c_1 ranges over all concepts in $s(x1)$ and c_2 over $s(x2)$, respectively.
While most approaches in this field take the taxonomy link distance into account (see: path-length based approaches 3.2.3), Resnik's approach tries to minimize the role of network edges in the determination of the degree of similarity. In his technique he only uses edges to determine the LCS. This has its own drawbacks. One disadvantage is a indistinguishably of any two pairs of concepts having the same LCS. For example, Figure 3.7 depicts that the similarity of $Sim_1(money, credit) = Sim_1(dime, creditcard)$, since the LCS in both cases is the *medium of exchange*. Path-length approaches try to expunge this problem.

Lin's Universal Similarity Measure
Resnik's measure depends highly on a taxonomy, such as WordNet. In contrast to this, Lin tried to abandon the dependencies to any form of knowledge representation. Yet still his measure represents a theoretically justified similarity measure for arbitrary objects. His basic intuitions are that the similarity of an object A and B is directly related to their *commonality*. The more commonality the two objects share, the more similar they are. The second foundation is related to the differences between two objects. The more

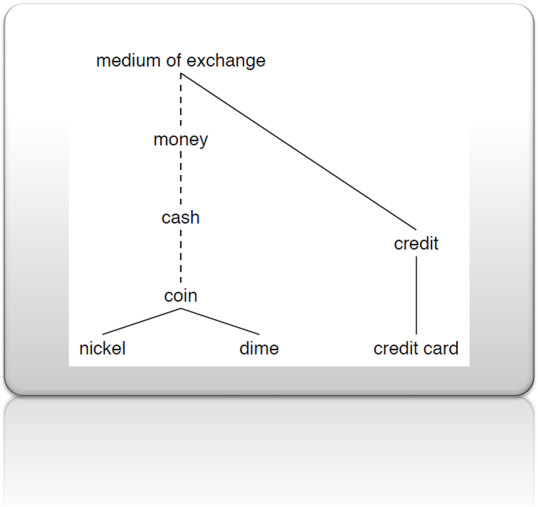


Figure 3.7: Fragment of the WordNet Taxonomy; Solid lines present hypernomy relations, Dashed lines indicate that some intervening nodes have been omitted. Adapted from [Res95].

differences A has to B the less similar they are. And finally, if A and B are identical objects, then the maximum similarity must be reached no matter how much commonality they share. Lin defined commonality as a measure of information content:

$$IC\left(comm(A,B)\right) \tag{3.28}$$

and the difference between objects as:

$$IC\left(descr(A,B)\right) - IC\left(comm(A,B)\right) \tag{3.29}$$

where $descr(A,B)$ are propositions describing what A and B is and $comm(A,B)$ describes the common features of A and B.
Based on these two definitions/assumptions Lin raises his similarity theorem:

$$Sim_2(A,B) = \frac{\log P(comm(A,B))}{\log P(descr(A,B))} \tag{3.30}$$

In the case that a similarity between two concepts should be measured, as in WordNet similarity approaches, the theorem in Equation (3.30) can be transformed into the similarity measure shown in Equation (3.31):

$$S_{15}(D_{x1}, D_{x2}) = \frac{2 * \log P(LCS(c_1,c_2))}{\log P(c_1) + \log P(c_2)} \tag{3.31}$$

where c_1 ranges over all possible concepts for the word $x1$ and c_2 analog.

Taxonomic Path Length

While information content based measures, such as Resnik (described in Section 3.2.3) or Lin (described in Section 3.2.3), are using node-count calculations for their similarity measures, edge-counting based measures are building a function around the number of nodes separating one another.

Edge-based similarity measures were initially presented by Rada et. al in 1989 on the MeSH (Medical Subject Headings) topology [RMBB89]. The MeSH topology is a semantic hierarchy for indexing articles in the bibliographic retrieval system Medline. The network contains 15.000 terms in a nine-level hierarchy, which are structured with a BROADER-THAN relation. Rada et al. followed the basic assumption that "the number of edges between terms in the MeSH hierarchy is a measure of conceptual distance between terms" [RMBB89].

Jarmasz and Szpakowicz [JS03] achieved equally good results as Rada et al. on the *Roget's Thesaurus* by treating the thesaurus as a simple hierarchy of clusters and computing the shortest path between two words within the clusters.

Hirst and St-Onge Path Length

In 1998, Hirst and St-Onge adapted the semantic distance based algorithm from the Roget's Thesaurus to WordNet [HSO98]. They classified the word-to-word relations into two categories: *strongly related* and *medium-strong related*.

Their fundamental idea was that two words are strongly related whenever they are in the same synset (i.e. *human* and *person*), they are in two different antonomy-connected synsets (i.e. *predecessor* and *successor*), or one of the words is a compound (or a phrase) that includes the other word (i.e. *school* and *private school*).

Medium strong relations are defined by Hirst and St-Onge if they are connected via a allowable path associated with each word. A path is allowable if it contains not more than five links and conforms to one of eight patterns depicted by Figure 3.8.

Summarizingly, Hirst and St-Orge's similarity function is denoted in Equation (3.32):

$$Sim_3(c_1, c_2) = C - len(c_1, c_2) - k * turns(c_1, c_2) \tag{3.32}$$

where $C = 8$ and $k = 1$ are empirically chosen constants and $turns(c_1, c_2)$ denotes the number of times the path between c_1 and c_2 changes its direction.

Sussna's Depth-Relative Scaling

Sussna's depth-relative scaling, presented in [Sus97], is based on the observation that sibling-concepts deep in the taxonomy appear to be more closely related than the one higher up. Thus, he weighted the existing *hypernym, hyponym, holonym*, and *meronym*

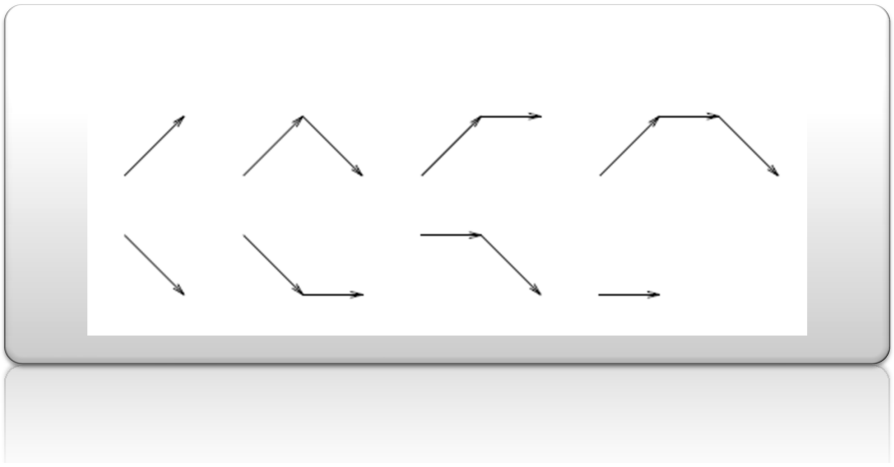


Figure 3.8: Patterns of Allowable Paths in Hirst and St-Orge's Medium-strong Concept Relations [HSO98].

edges/relations in the WordNet noun hierarchy. Each weight corresponds to the inverse-relation that contains either a weight or a range of weights $[min_r, max_r]$. The weights $wt(c, r)$ of the type r from some concept c is reduced by a factor that depends on the number of edges ($edges_r$) of the same type leaving c_1 (see: Equation (3.33)):

$$wt(c_1, r) = max_r - \frac{max_r - min_r}{edges_r(c_1)} \tag{3.33}$$

As an intermediate result, Sussna defines the distance between two *adjacent* concepts as the average of the weights in each direction (i.e. r' being the inverse of r) as depicted in Equation (3.34):

$$Sim_4^{adj}(c_1, c_2) = \frac{wt(c_1, r) + wt(c_2, r')}{2 * max[depth(c_1), depth(c_2)]} \tag{3.34}$$

where r and r' is the relation between c_1 and c_2, respectively its inverse.
As a final result Sussna defines the semantic relatedness between two arbitrary concepts as the sum of distances between pairs of adjacent nodes along the shortest path (see: Equation (3.35):

$$Sim_4(c_1, c_2) = \sum_{c_i, c_j \in Path(c_1, c_2)} Sim_4^{adj}(c_i, c_j) \tag{3.35}$$

where $Path(c_1, c_2)$ denotes the shortest path between c_1 and c_2 and c_i, respectively c_j are adjacent nodes along the path.

Wu and Palmer's Conceptual Similarity

Wu and Palmer propose in [WP94] a *conceptual similarity*. It is derived from the verbal concept similarity in the projected domain hierarchy when translating from English verbs to Mandarin Chinese. According to Wu and Palmer, the similarity of two words is the scaled sum of all their senses compared to each other (see Equation (3.36)):

$$Sim_5(c_1, c_2) = \frac{2 * depth(LCS(c_1, c_2))}{len(c_1, LCS(c_1, c_2)) + len(c_2, LCS(c_1, c_2)) + 2 * depth(LCS(c_1, c_2))} \tag{3.36}$$

where $depth(LCS(c_1, c_2))$ is the depth of the least common subsumer of c_1 and c_2. Figure 3.9 represents this circumstance visually.

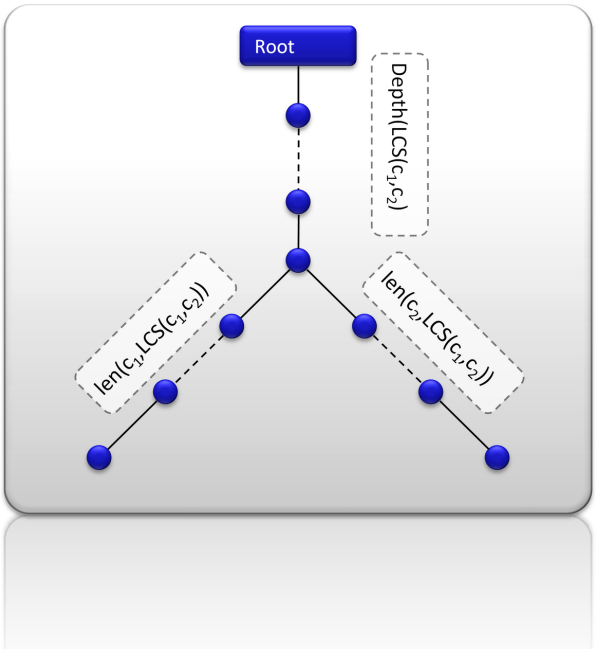


Figure 3.9: Path-based Conceptual Similarity of Wu and Palmers; Adapted from [WP94].

As the final step, Wu and Palmer define the similarity between two word's meanings as in Equation (3.37)

$$S_{16}(D_{x1}, D_{x2}) = \sum_i w_i * Sim_5(c_{i,x1}, c_{i,x2}) \tag{3.37}$$

where w_i is the weight of each pair of concepts in each domain (the sum of all w is 1) and i iterates over all possible word to concept assignments.

This model is appropriate for measuring the similarity of both verbs and nouns in every Is-A hierarchy.

Leacock and Chodorow's Normalized Path Length
Leacock and Chodorow suggest in [LC98] a scaled semantic similarity that is similar to Wu and Palmer's approach (see: Section 3.2.3). It is defined as follows in Equation (3.38):

$$Sim_6(c_1, c_2) = -\log \frac{len(c_1, c_2)}{2 * maxDepth(WordNet)} \tag{3.38}$$

where $maxDepth(WordNet)$ is the maximum depth in the given hierarchy and $len(c_1, c_2)$ is the shortest path between the two concepts.

Jiang and Conrath's Combined Approach
Jiang and Conrath [JC97] combined the edge-based approach with a node-based technique that is known from the information content calculation. They assumed that a combination of information content and edge-counting will improve the correlation coefficient. Thus, they considered the link type, depth, conceptual density, and information content of the compared concepts.
In a hypernym/hyponym framework, such as WordNet's noun hierarchy, Jiang and Conrath postulate that the semantic distance of the link connecting a child-concept c to its parent-concept $parent(c)$ is proportional to the conditional probability $P(c \mid parent(c))$ of encountering an instance of c given an instance of $parent(c)$ [JC97]. In a specific formula:

$$Sim_7(c, parent(c)) = -\log\left(P(c \mid parent(c))\right) \tag{3.39}$$

However, by definition $P(c \mid parent(c))$ can be rewritten to the following Formula (3.40), by taking advantage of Resnik's information content formulation (see: Section 3.2.3):

$$P(c \mid parent(c)) = \frac{P(c \cap parent(c))}{P(parent(c))} = \frac{P(c)}{P(parent(c))} \tag{3.40}$$

The second part of Equation (3.40) can be applied due to Resnik's assigning of concept probabilities (more information is given in [Res95]), since a child concept is automatically an instance of its parent. Furthermore, analog to Resnik's definition of information content one can see that:

$$Sim_8(c, parent(c)) = IC(c) - IC(parent(c)) \tag{3.41}$$

Given the formula between adjacent nodes in Equation (3.41), the semantic measure between arbitrary pairs of nodes can be computed according to the common practice, which is to take the sum of the shortest path connecting the two concept nodes. However, this distance measure can be rewritten to a more appropriate form, which is the final similarity measure of Jiang and Conrath. The formula is depicted in Equation (3.42):

$$\begin{aligned} Sim_9(c_1, c_2) &= IC(c_1) + IC(c_2) - 2 * IC(LCS(c_1, c_2)) \\ &= 2\log(P(LCS(c_1, c_2)) - (\log(P(c_1) + \log(P(c_2))))) \end{aligned} \tag{3.42}$$

WWW-based measures

This section will give an insight into the approaches, which abandon a classical structured knowledge repository and switch over to the World-Wide-Web as their relation information source.

Google Similarity Measure

Cilibrasi and Vitányi present in [CV07] the *Google Similarity Measure*. Unlike the latter mentioned approaches, which depend on some structured knowledge source (i.e. the WordNet taxonomy), this technique builds upon the world-wide-web (WWW) as a source of meaningful knowledge. Cilibrasi and Vitányi note that the WWW presents "[...] amorphous low-grade knowledge available for free [...]", which is "[...] typed in by local users aiming at personal gratification of diverse objectives, and yet globally achieving what is effectively the largest semantic electronic database in the world" [CV07]. Thus, the WWW can be seen as the largest database on earth, and its context information entered by millions of independent users averages out to provide useful semantics. Accessed by an information retrieval engine, such as the *Google search engine*, this knowledge source can return aggregate page-count estimates for a large range of search queries. Cilibrasi and Vitányi base their similarity measure on two major theories: *Information distance* and the *Kolmogorov complexity*. The information distance, whose theory and schema is described by Bennett et al. in [BGG+98], constructs an imaginable *binary program* with the following properties: Given two strings x and y, the length of the shortest binary program in a *reference universal computing system*, which computes from the input x the output y and additionally from the input y the output x, is called the *information distance* denoted by $E(x, y)$.
The other major theory is the *Kolmogorov complexity* (see: [LV97] for more information). The Kolmogorov complexity of a string x (denoted by $K(x)$) is defined as the length, in bits, of the shortest computer program of a fixed reference computing system that produces the output x. One way to think about the Kolmogorov complexity $K(x)$ is to view it as the length, in bits, of the ultimate compressed version from which x can be recovered by a general decompression program [CV07]. Compressing x by an arbitrary

compressor, such as *gzip*, *bzip2*, or *PPMZ*, results in the compressed version x_{compr} with the property $|x| < |x_{compr}|$. Hence, every compressor can be seen as an approach to reach the Kolmogorov complexity $K(x)$ which retrieves the ultimate lower bound compression version of x.
Combining the information distance with the Kolmogorov complexity yields to the following circumstance:

$$E(x,y) = K(x,y) - \min\{K(x), K(y)\} \tag{3.43}$$

where $K(x,y)$ is the length of the shortest program that produces the pair x, y. One can prove that $E(x,y)$ is actually a metric. However, since $E(x,y)$ is theoretical and not computable one has to approximate the information distance by using Equation (3.44):

$$E(x,y) \leq D(x,y) + c_D \tag{3.44}$$

where $D(x,y)$ is a computable approximation of the information distance and c_D is an error-factor dependent on the chosen approximation function (e.g. a compressor).
Although it can be proven that $E(x,y)$ is the universal information distance of two strings x and y (see: [CV07] for more information) it is still not normalized to the input's size. If small strings differ by a large information distance in comparison to their sizes, then the strings are understood to be different. On the other hand, if two very large strings differ by the same information distance, which is in this case a relatively small amount, then the two strings are very similar. As a consequence, the information distance itself is not suitable to express true similarity and suffers from a length-based normalization. This normalization can be given by the dominator, which construes the maximum of the Kolmogorov of x and y given by the following Formula (3.45):

$$NID(x,y) = \frac{K(x,y) - \min\{K(x), K(y)\}}{\max\{K(x), K(y)\}} \tag{3.45}$$

Cilibrasi and Vitányi proved that is eligible to use the Google machinery for approximating the absolute probabilities for the information content and suggest to use the number of hits from a search engine as an approximate to the Kolmogorov in the *Normalized Google Distance*:

$$\begin{aligned} NGD(x,y) &= \frac{G(x,y) - \min\{G(x), G(y)\}}{\max\{G(x), G(y)\}} \\ &= \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log N - \min\{\log f(x), \log f(y)\}} \end{aligned} \tag{3.46}$$

where $f(x)$ denotes the number of pages containing x, $f(x,y)$ depicts the hits for both

query words x and y, and N is a normalization factor typically chosen to be larger than the overall amount of pages indexed by Google ($8 * 10^9 \leq |Google_{Index}| \leq 9 * 10^9 \leq N$). Google's similarity measure has the advantage that it is easy to implement and accesses the largest possible information source of the world. Thus, WWW-corpus based similarity measures it can be seen as the only representatives that do not have an inherent domain dependency, resulting from a limited/domain-dependent corpus. Despite its ease of implementation the Normalized Google Distance has functional disadvantages, such as Google's restriction to at most 500 search queries per IP and day.

Pointwise Mutual Information - Information Retrieval
In [Tur01], Turney presents an approach for recognizing synonyms, which can be used for measuring semantical similarity. His statistical measure, called *Pointwise Mutual Information - Information Retrieval*, short PMI-IR, also uses the results of an online information retrieval mechanism to compute the relatedness between words or phrases. Turney uses word co-occurrence counts collected from a large corpus –in his example the WWW. His measure is showcased in Equation (3.47):

$$\text{PMI-IR}(D_{x1}, D_{x2}) = \log_2 \frac{P(x1 \cup x2)}{P(x1) * P(x2)} \tag{3.47}$$

where $x1$ and $x2$ are words, $P(x1 \cup x2)$ relates to the probability of a co-occurrence of $x1$ and $x2$, and $P(x1)$, respectively $P(x2)$ are the probabilities of encountering the words separately. The numerator of the formula, $P(x1 \cup x2)$, measures the statistical dependence (co-occurrence dependency). If the terms are likely to be related, then the numerator will have a value greater than the denominator, which represents the independence assumption. Measuring the logarithm-weighted fraction of dependence/independence leads to the term *pointwise mutual information*.
In order to get access to the largest knowledge source, Turney suggest to make effectively use of the *Altavista* search engine [Alt11]. In comparison to other search engines, Altavista is more suited to handle this problem, because it offers the *Altavista Advanced Search Query Syntax*. This query syntax allows the users to issue Near queries returning the co-occurrence within a ten-word window. By this mean Turney approximates $P(x1 \cup x2)$ with the following Equation (3.48):

$$P(x1 \cup x2) \approx \frac{hits(near(x1, x2))}{Websize} \tag{3.48}$$

With the assumption that $P(x_i)$ can be approximated by $P(x_i) = hits(x_i)/Websize$ the final PMI-IR formula can be obtained by Equation (3.49):

$$\text{PMI-IR}(D_{x1}, D_{x2}) = \log_2 \frac{hits(and(x1, x2)) * Websize}{hits(x1) * hits(x2)} \tag{3.49}$$

where $Websize$ is approximated with Chklovski's co-occurrence experiment's Websize approximation: $7 * 10^{11}$ [CP04].

3.2.4 Combined Semantic and Syntactic Measures

Liu's Semantic Dynamic Time Warping Approach

While most approaches in the semantic similarity field try to assess the similarity between concepts in an accurate manner, Liu faces the problem of calculating the semantic similarity between sentences. His approach takes the concept semantic similarity, word order and the contribution of different parts of speech into account.

Measuring the similarity between concepts is feasible and proven, as we show above. However, finding the maximum alignment of matches from one sentence's concepts to the other sentence's concepts is $\mathcal{NP}$-hard. This is due to the reason that a combinatorial explosion exists in the possible pairwise alignments. Liu faces this problem with a *Dynamic Time Warping* (DTW) technique. He divides the sentence into two subsequences, according to their specific part-of-speech (POS) tags, and calculates the similarity as a weighted combination of these two similarities. This differentiation incorporates the assumption that some POS –noun phrases and verbs– contribute more to the semantical information in a sentence, than other POS –adjectives and adverbs.

Liu's main pillar is the concept similarity measure, which resembles other path-based concept similarity measures (i.e. Wu and Palmer's Conceptual Similarity see: Section 3.2.3 or Hirst and St-Onge's Path Length Similarity see: Section 3.2.3), is the following concept similarity function:

$$Sim_{10}(c_1, c_2) = \frac{f(depth(LCS(c_1, c_2)))}{f(depth(LCS(c_1, c_2))) + f(shortestPath(c_1, c_2))} \tag{3.50}$$

where f is an arbitrary transfer function (Liu suggests $f(x) = \exp^x - 1$).

In order to align different sentences to each other, Liu transforms the sentence into a sequence of words $Seq_1 = < w_1^1, w_2^1, \ldots, w_m^1 >$ with length m and the other sentence into $Seq_2 = < w_1^2, w_2^2, \ldots, w_n^2 >$ with length n. Subsequently, he constructs a $m \times n$ grid, where each grid element (i, j) represents the alignment of words/concepts w_i^1 and w_j^2. The task is then to find a minimal warping path through the grid beginning from the bottom-most right element:

$$D_{dtw}(D_x, D_y) = f(m, n) \tag{3.51}$$

$$f(i, j) = (1 - Sim_{10}(concept(w_i^1), concept(w_j^2))) + \min \begin{cases} f(i-1, j) \\ f(i, j-1) \\ f(i-1, j-1) \end{cases} \tag{3.52}$$

where $f(0, 0) = 0$, $f(i, 0) = f(0, j) = \infty$, $i \in (0, m)$ and $j \in (0, n)$.

The DTW-alignment D_{dtw} will lastly be normalized by the warping path length t as depicted in Equation (3.53):

$$D_{seq}(D_x, D_y) = D_{dtw}(D_x, D_y)/t \tag{3.53}$$

Finally, having the sequence alignment formula Liu retains the sentence's word ordering with the help of the following scheme:

$Sen =$ "The lovely baby stopped crying, when it saw the red apple"
$Seq_{nv} =$ <baby, stopped, crying, saw, apple>
$Seq_{aa} =$ <the, lovely, when, it, the, red>

which draws out nouns and verbs from Sen to get Seq_{nv} and adjectives and adverbs to get Seq_{aa}. Note that the word ordering is still retained in this schema, which enables the DTW technique to incorporate word ordering into the similarity measure. The final Liu Semantic Dynamic Time Warping similarity measure is depicted in Equation (3.54):

$$S_{17}(D_x, D_y) = \gamma * D_{seq}(Seq^1_{nv}, Seq^2_{nv}) + (1 - \gamma) * D_{seq}(Seq^1_{aa}, Seq^2_{aa}) \tag{3.54}$$

3.3 Text Summarization

When scanning through a news aggregator's overview page the users base their decision whether or not to read an article on the headline and the article's description text. This description text is usually an fixed-size excerpt of one news' full text. While it originates in most cases from the most recent article of the topic, retrieving it from the most relevant article leads to better results. Yet still, the best solution is to present a meaningful text excerpt produced from the entire news cluster. The technique behind this is called *extractive multi-document text summarization*. Hovy defines in [LH03]: "A summary is a text that is produced from one or more texts that contain a significant portion of the information in the original text(s), and that is no longer than half of the original text(s)."
The text summarization literature can be subdivided into *abstractive* and *extractive* approaches. Extractive text summarization produces a summary of text copied from the source material. Typically, such approaches are based on shallow analysis, as Figure 3.10 presents. Abstractive text summarization approaches, on the other hand, produce a textual summary that is usually based on deeper analysis. Many systems in the text summarization field employ hybrid methods.

In general, extractive systems can be characterized as "knowledge-poor", which is contrasted against "knowledge-rich" abstractive approaches. Knowledge-rich abstractive

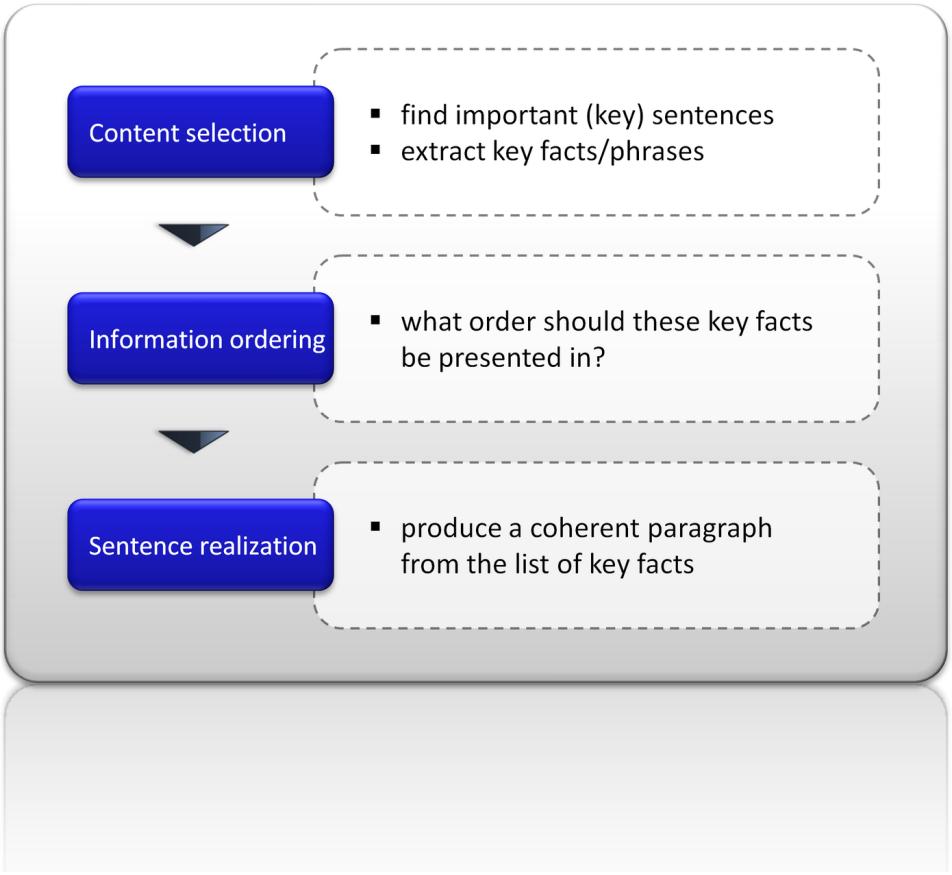


Figure 3.10: Extractive Text Summarization.

techniques involve one or more of the following: detailed linguistic analysis on the source text to produce richly-annotated structures, incorporation of world knowledge to support the transformation process, or generation of fluent natural language text from abstract representations [JL09].
Most research projects in the summarization field are primarily extractive. As a general pipeline these techniques first segment source text into smaller units, such as sentences, paragraphs and then score these units according to a variety of features. Examples of such features are the position in the text [Edm69], term and phrase frequencies [Luh58], lexical chains (degree of lexical-connectedness between various segments) [BE97], topics present in the text [Rad01], or discourse prominence [Mar97]. A widely-adopted approach is to use machine learning techniques to determine the relative importance of various features (one of the earliest example being [KPC95]) (adapted from [JL09]).

In the following several techniques to produce extractive summaries will be presented. Each of the presented approaches is outstanding from the large research

publication sphere, because they have unique approaches to each of the pipeline steps in Figure 3.10.

3.3.1 MEAD

The University of Michigan's summarization system, named *MEAD* [RJST04], was developed to produce multi-document extractive summaries. Its intuitive idea is to cluster all documents in a corpus to retrieve all available topics. These topics are represented by a centroid-based method. A centroid is defined as a set of words that are statistically important to a cluster of documents. Following this intuition, centroids identify salient words or sentences in a cluster. Whether or not a document is similar to the cluster centroid is computed with a modified cosine similarity measure (see also: Section 3.2.1), as depicted in Equation (3.55) and described in detail in [RHM99].

$$S_{18}(D,C) = \frac{\sum_k (tf_{d,k} * tf_{c,k} * idf(k))}{\sqrt{\sum_k (tf_{d,k})^2}\sqrt{\sum_k (tf_{c,k})^2}} \tag{3.55}$$

where the index k denotes the k^{th} word in a document d, respectively cluster-centroid c. The formula takes the discriminability of a term (depicted as $idf(k)$) into account. Thus, salient terms can be promoted, whereas often occurring terms in the corpus are punished.
In order to produce an extractive summary, the MEAD extraction algorithm decides which sentences to include in the extract by ranking them in accordance to a set of features.

Centroid Value

The first feature for determine the saliency of a sentence is called the *centroid value*. It is computed according to the following Equation (3.56):

$$C_i = \sum_w C_{w,i} \tag{3.56}$$

The centroid value C_i for a sentence S_i is thereby computed as the sum of all centroid values $C_{w,i}$ for all words in the sentence.

Positional Value

The positional value takes the ordering of sentences within a document into account. This follows the hypothesis that sentences in the beginning of a document are more important than the once in the latter parts. It can be computed according to Equation (3.57):

$$P_i = \frac{(n - i + 1)}{n} * C_{max} \tag{3.57}$$

First Sentence Overlap
An overlap value is taken into account, too. It punishes sentences that are too close the other already chosen sentences. In order to avoid computational complexity, this formula dismisses all but the first sentence of the investigated document. It is presented by Radev in [RJST04] as:

$$F_i = \vec{S_1}\vec{S_i} \tag{3.58}$$

Feature Combination
Radev combines all mentioned features as a weighted combination:

$$Score(S_i) = w_c C_i + w_p P_i + w_f F_i - w_R R_s \tag{3.59}$$

where w_c, w_p, w_f and w_R are empirically chosen weights and the negative factor R_s presents a redundancy penalty for too similar sentences.

3.3.2 DEMS and MultiGen

Schiffman et al. present in [SNMS02] a further approach towards extractive multi-document summarization. Their solutions, called Dissimilarity Engine for Multidocument Summarization, short DEMS, selects informative and interesting sentences based on the *concept frequencies* of the sentence. It is used in tandem with the MultiGen summarization engine [BME99]. Both solutions are implemented in the Columbia NewsBlaster system (see: Section 2.4 "Related Work")
The MultiGen summarization engine is specifically designed for news articles presenting different descriptions of the same event. Since hundreds of news stories on the same event are produced on a daily basis by news agencies, this repeated information may be seen as a good indicator of its importance and can thus be used for the summarization task.
Yet still, extracting all similar sentences from a news cluster would produce a verbose and repetitive summary. Hence, MultiGen and DEMS move beyond sentence extraction. They facilitate a comparison of the extracted similar sentences to select the phrases that should be included in the summary. Moreover they reformulate the phrases as new text with the help of sentence generation engine [BME99].
Their model is combined of three inter-operating components, called *planners*. First, the content planner selects and orders propositions from an underlying knowledge base to form the textual content. Second, a sentence planner combines the propositions into

a single sentence and the sentence generator realizes each set of combined propositions as a sentence, mapping from concepts to words and taking care of syntactic sentence structures.

Content Selection

The content selection follows the idea that non-identical phrases can still report the same fact. Therefore, sentences are compared according their predicate-argument structure, which is produced for each sentence by a dependency parser. Dependency parsers represent a sentence as the dependency between constituents, while ignoring irrelevant features, such as constituent ordering. The resulting constituent tree naturally represents predicate-argument structures. Each non-auxiliary word in the sentence has a node in the dependency parser tree, and these nodes are connected to its direct dependent. Figure 3.11 shows a sample sentence and its dependency tree:

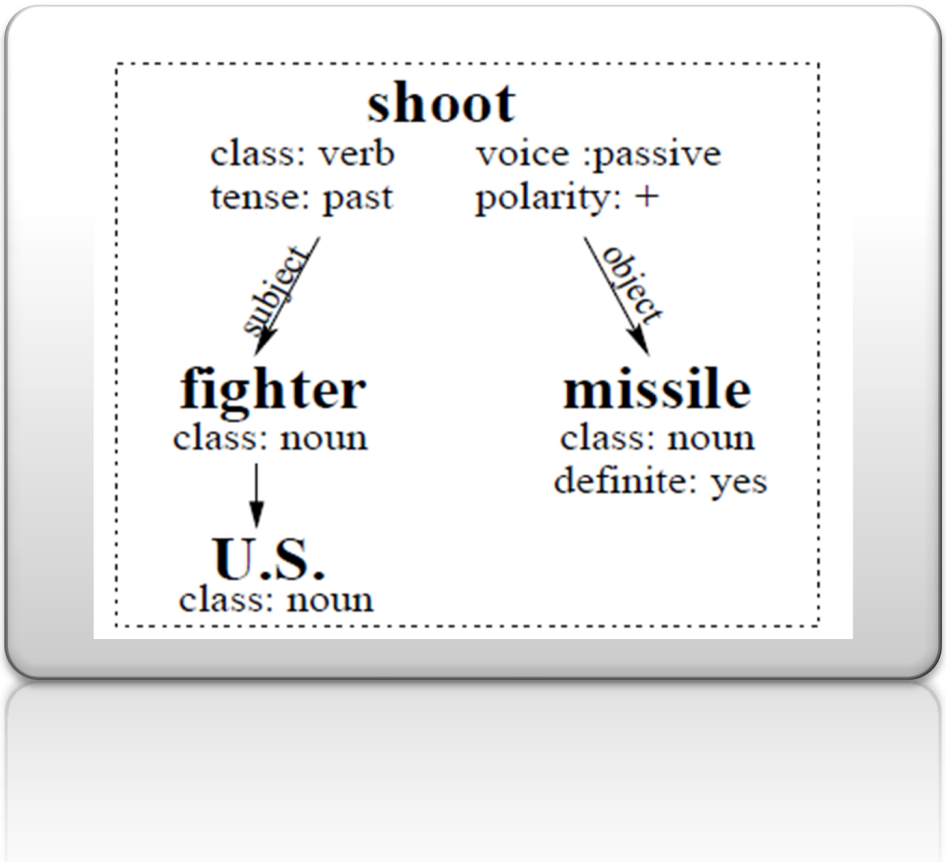


Figure 3.11: Dependency Tree for the Exemplary Sentence "*U.S. fighter was shot by missile.*" [BME99].

In order to select important content these dependency parser trees are compared. Starting from the root node two trees are traversed recursively with the following schema: If two nodes are identical, they are added to the output tree and their children are compared. Once a full phrase has been found, it is added to a theme intersection list. If the nodes are not identical an approximate corpus-induced paraphrase rule matching scheme helps to find further theme intersections.

Sentence Planner and Generator
The sentence planner takes the original document's temporal sequence of events into account. Despite that, the more interesting part is the sentence generation. Its input is the shallow parsing output, described in the latter "Content Selection" part (see: Section 3.3.2). This dependency tree reveals important information, such as semantic roles, temporal descriptions or sources. Transforming it again into a textual representation requires the definition of a grammar specification language, which functions as a mapping from the predicate-argument structure produced by the content-planner to the functional representation expected by a formalism interpreter and text generator. After specifying such a grammar, special natural language generators, in this case FUF and SURGE ([Kha99], respectively [ER98], can be used for the final natural language summary generation.

3.3.3 LexRank

Erkan and Radev present in [ER04] a promising stochastic graph-based method for computing the relative importance of text units, called *LexRank*. Although most text summarization approaches are basing their salience score on a centroid-based similarity measure, LexRank considers a new approach for computing a sentence's importance. It is based on the concept of eigenvector centrality in a graph representation of sentences. In the graph theory, a common assumption is that the nodes which are highly connected are likely to carry more salient information than others.

LexRank bases its salience score on the graph theoretical concept of prestige as it was suggested by Brin and Page in their famous *PageRank* algorithm [BP98]. PageRank, which is the foundation of the Google search engine ranking, follows the intuition that a website's relative importance can be traced back to the relative importance of all websites that link to it. Therefore, PageRank considers recursively the importance of each page that casts a vote (has a hyperlink to the document).
Alike the mapping from websites to nodes, LexRank assigns text units to nodes and the intra-unit similarity to the edges connecting the nodes. In their paper [ER04] the authors suggest to use a modified cosine similarity, as it is shown in Equation (3.55). However, every possible similarity measure can be used to assess the text unit's

similarity. With the help of these values, LexRank builds a similarity matrix and interprets the similarity scores as edge-weights in a graph. This fully-connected graph can be condensed to a partially-connected graph by discarding edges below a chosen threshold, as depicted in Figure 3.12.

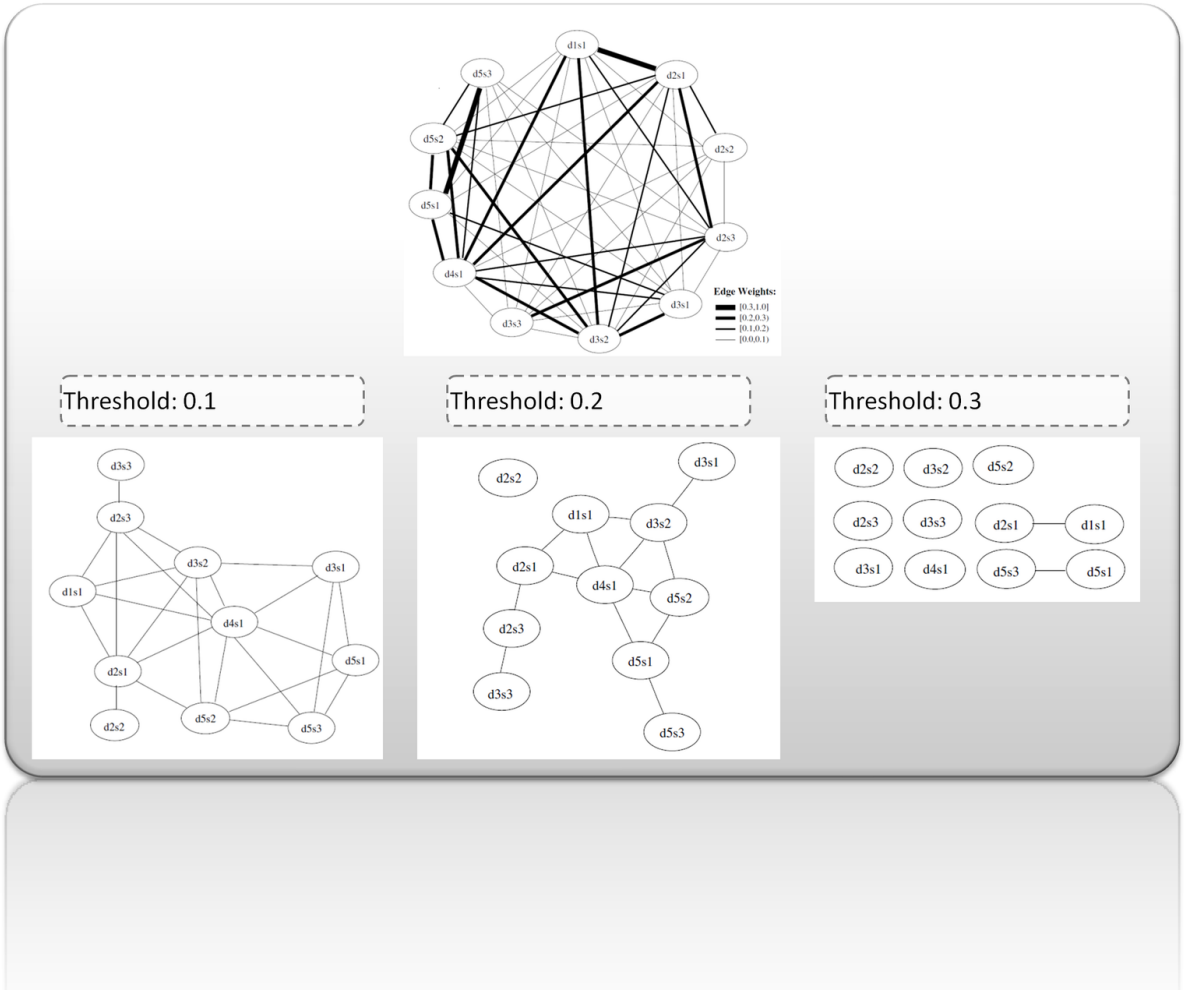


Figure 3.12: LexRank Similarity Graphs with different Edge-Removal Thresholds; Adapted from [ER04].

Discarding edges in a graph-based directly corresponds to eliminating less significant similarities. Too low thresholds may mistakenly take weak similarities into consideration while too high thresholds may lose many of the similarity relations in a cluster. In order to compute a node's importance in the graph a simple definition refers back to the notion of degree. The higher a node's degree the more salient it is. This interpretation of centrality is called *degree centrality*.
Degree centrality may have a negative effect in the quality of the summaries in some

cases where several unwanted sentences vote for each other and raise their centrality. To avoid this situation one can consider where the votes come from and take the centrality of the voting nodes into account. Alike in the PageRank example, the idea is to consider every node having a centrality value and distributing this centrality to its neighbors. This formulation can be expressed by the Equation (3.60):

$$p(u) = \sum_{v \in adj[u]} \frac{p(v)}{deg(v)} \tag{3.60}$$

where $p(u)$ is the centrality of node u, $adj[u]$ returns adjacent and connected nodes to u and $deg(u)$ returns the degree of node u. In accordance to this definition one can rewrite Equation (3.60) into the matrix notation:

$$\mathbf{p} = \mathbf{B}^T \mathbf{p} \tag{3.61}$$

where the matrix $\mathbf{B}$ is derived from the adjacency matrix of the similarity graph by dividing each element with the corresponding row sum.
In order to guarantee that the eigenvector $\mathbf{p}$ exists and can be uniquely identified, Erkan and Radev prove that the matrix $\mathbf{B}$ is a stochastic matrix (for more information see [ER04] and [BP98]), which can be assured to be *irreducible* and *aperiodic*. In dependence on, Page et al. they suggest to reserve a low probability for jumping to an arbitrary node in each step of the produced Markov chain. Hence, Equation (3.60) has to be modified to:

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in adj[u]} \frac{p(v)}{deg(v)} \tag{3.62}$$

or in the matrix notation:

$$\mathbf{p} = [d\mathbf{U} + (1-d)\mathbf{B}]^T \mathbf{p} \tag{3.63}$$

The proposed method from Erkan and Radev has several advantages over the classical centroid-based methods. First and most importantly, subsumption among sentences is implemented in the idea. If one sentence's information content comprised a different sentence's information content, then the first will be preferred. This is due to the fact that the first node's degree centrality will be leveraged positively by the second sentence. The other advantage of LexRank is that the methodology prevents high Idf scores from boosting up the score of a sentence that is unrelated to the topic. In centroid-based methods, the damping or boosting Idf scores have a significant impact on the centroid calculation. Hence they drive the importance of rare terms even if they do not occur elsewhere in the cluster.

3.4 Relevance Feedback

Reading news articles on the web site of a news aggregator inevitably links to the news organization's web site on which the content is actually published. This is due to the fact that the data control needs to be in the news organization's hands in order to earn money from advertising contracts.
In contrast to the practiced circumstance in the web sphere, (offline-) news exploration systems can significantly improve the reading experience if they incorporate *relevance feedback* mechanisms. Its main concept is that a search engine presents the user a set of search results and the user assesses their relevance to the search topic. In an iterative process this *relevance information* is sent back and forth between the search engine and the user to produce a new query or expand the original one.

Most generally relevance feedback can be divided into four subcategories, which will be described in the following sections:

3.4.1 Conventional Relevance Feedback

Standard user-driven conventional relevance feedback can be implemented into client-server architectures for web sites, as well as in stand-alone desktop applications. Most of the web-based mechanisms can be directly transformed onto the desktop applications and vice versa. In order to avoid redundancies this section will only highlight web-based conventional relevance feedback mechanisms.
User interface relevance feedback mechanisms, such as textual input fields, checkboxes and menu choices can be directly implemented in client-server architectures using the Hypertext Markup Language (HTML). This gives users the chance to submit relevancy decisions back over the Hypertext Transfer Protocol (HTTP) to the search engine/content provider. The relevance feedback mechanism provided by the standard HTTP/HTML setting is mostly limited to simple "key equals value" pairs, where the values are typically Boolean values. This starting point for relevance feedback is called *conventional relevance feedback*.

3.4.2 Implicit Relevance Feedback

The term *implicit relevance feedback* refers to the intentionality of the user's behavior. As Figure 3.13 depicts it incorporates mechanisms, such as eye movements, gestures and speech. This enumeration can further be enlarged to the ordinary selection of an item. If the user –in the news exploration example– selects an outstanding article, these features can guide the subsequent news reading process. For instance, a user that clicks on an image could be provided with additional image material related to the article.

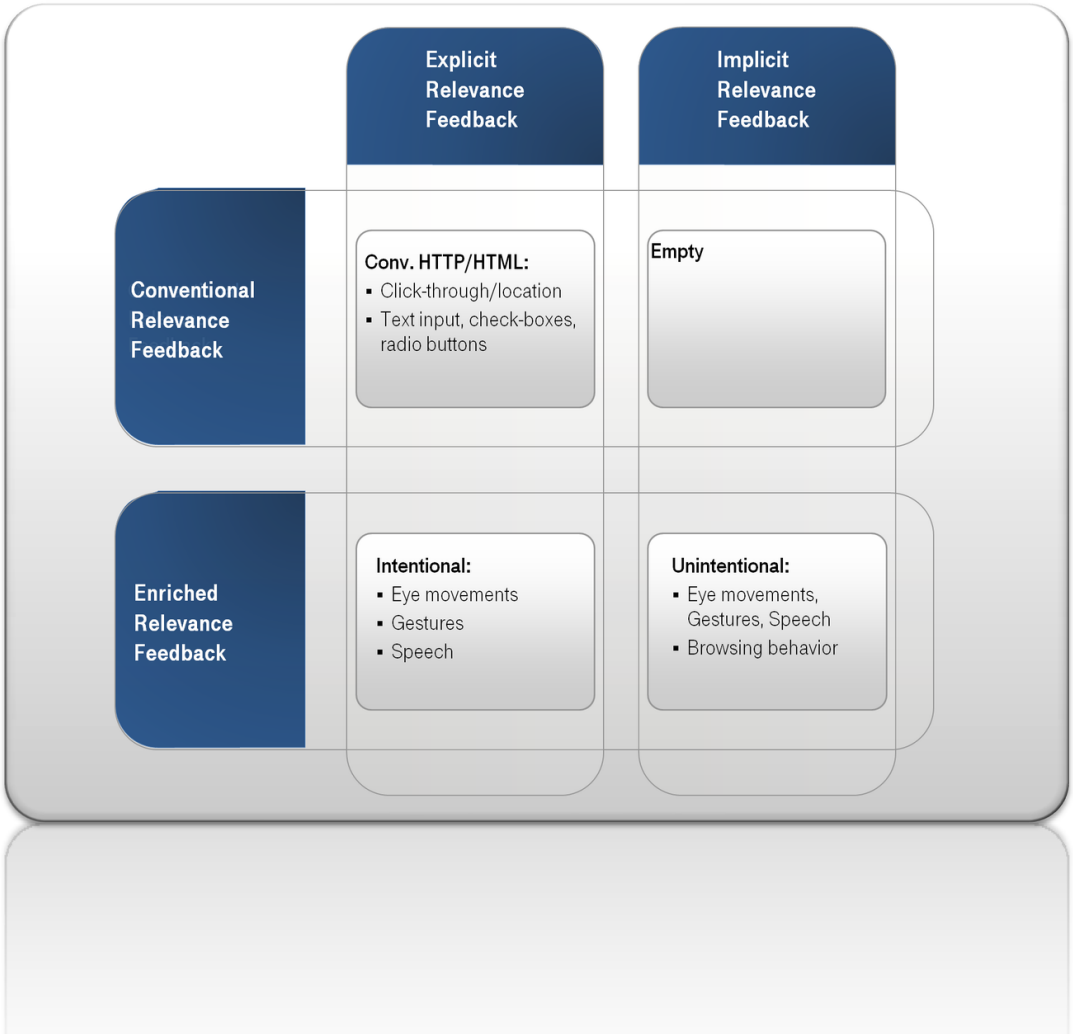


Figure 3.13: Relevance Feedback Subcategories [ZKL08].

If the human-computer interface is able to record the unintentional order of the consecutive checkbox clicks, their timing and the path the mouse pointer traverses between the clicks, then this data can be regarded as implicit relevance feedback information.
One special case of implicit relevance feedback systems are gaze-based interfaces. These systems track the user's eye-movement and allow retrieving points of focus for the investigated interface. However not only the focus points are of interest for the relevance feedback. Rather so, the eye's rest times give a direct hint for the concentration and conscious process of gathering information.

3.4.3 Enriched Relevance Feedback

Enriched relevance feedback denotes all improvements to the conventional relevance feedback that allow information retrieval mechanisms to suggest further content in a immersive way. More specifically, every feedback mechanism that improves boolean-valued relevance feedback implemented with basic HTTP/HTML forms counts to the subcategorization. Recent studies on the physiological behavior of subjects, such as in [RLC+06], point to the fact that unintentional feedback could be measured through the heart rate, eye movement rate, galvanic skin response, body temperature and the movements of the subject, too.
Other enriched relevance feedback methods could also be transferred to the news exploration domain. For example, behavioral profiles can be easily produced by analyzing event log files of registered users. For instance, a user who regularly visits the news category "Sports" and spends little to no time on "Politics" will be highly satisfied if his news portal adapts to his reading behavior.

For a further investigation of the field of relevance feedback the reader should be referred to the comprehensive report of Zhang et al. in [ZKL08].

3.5 Text Information Extraction

News exploration systems are a subform of *information retrieval systems*. As a consequence of this they rely on some form of indices or keywords, which are associated as meta-data to documents in the collection. The most important forms of keywords in news are related to the topic, the associated names (persons, organizations, places, etc.), and the geo-location.

3.5.1 Keyword Extraction

Keywords are descriptive words or phrases in the investigated article or document corpus. A standard algorithm of retrieving important keywords was supposed by Salton and Buckley in [SB88]:

Algorithm 3.4: Automatic Keyword Extraction Algorithm

```
1. Identify all individual words.
2. Use a dictionary of common high-frequency function words
   (i.e. stop word list) to ignore these words from the text.
3. Use a suffix stripping routine (a morphological analyzer) to reduce
   the words to their stems.
4. Remove all terms whose count is below a pre-specified threshold.
```

```
5.  Compute a term-weight for each word stem
6.  Represent a document as a vector of the weights of all index terms
    in a collection (i.e., all word stems found by this procedure).
```

The algorithm in Figure 3.4 remains vague in step #5 (term-weighting). In order to demonstrate the main approaches in this field, the following section will describe the mostly used technique for term weighting and its alternatives. These techniques are –to a great extend– part of every information retrieval system, short IR system.

Term Frequency - Inverse Document Frequency
The mostly used standard technique for the term-weight computation is the *Term Frequency - Inverse Document Frequency*, short Tf-Idf:

$$\text{Tf} - \text{Idf}(x, D_x) = tf_x * \log\left(\frac{N}{df_x}\right) \tag{3.64}$$

The weight for the term x is determined by the product of the number of occurrences of x in the document D_x, multiplied by the logarithm of the number of documents N divided by the number of documents in which the term occurs (df_x).
This weighting scheme yields to high values for terms, which are frequent in the specific document, but appear in only a few documents. Consequently, Tf-Idf can be seen as an indicator for the saliency of the term x in the document D_x.

Sublinear Term Frequency Scaling
A rarely used alternative/variation of Tf-Idf is the *sublinear term frequency scaling*. The Equation (3.65) shows its formula:

$$\text{Sublinear-Tf}(x, D_x) = \begin{cases} 1 + \log(tf_x), & \text{if } tf_x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.65}$$

Sublinear term frequency scaling follows the intuition that the term frequency is a measure of significance, but its effect has to be damped for high term-frequency values. As an example, a term with twenty occurrences in a document is considerable, yet still it is not twenty times more significant than a word that occurs one time.

Maximum Term Frequency Normalization
Maximum term frequency normalization scales the term frequency weights of all terms occurring in a document by the maximum term frequency in the specific document:

$$\text{Norm-Tf}(x, D_x) = a + (1 - a)\frac{tf_x}{tf_{max}} \tag{3.66}$$

where $tf_{max} = \max_{\tau \in D_x}(tf_\tau)$ and τ ranges over all terms in the document D_x. The constant a is empirically chosen value in the interval $[0, 1]$.
Norm-Tf's most important disadvantage is its sensitivity to outliers. If a document contains an outlier term with an unusually high term-frequency, it will outweigh all other terms in the document. Hence, the weight is far more representative for the document than it should be. Generally, raw term frequency based methods suffer from the fact that they do not incorporate any measure for the discriminability of a term (such as the inverse-document frequency Idf).

3.5.2 Named Entity Extraction

News describe a real or imaginary circumstance in the world. Consequently, the news' main entities are the protagonist and optionally the antagonist. Around these actors the story is woven. The task of the *Named Entity Extraction* is to identify the entities' names, which can be either names of people, places or names of organizations.
The ability to recognize previously unknown entities can be lead back to a classification problem. The classifier is trained by a distinctive modeling of features associated with positive and negative examples. After the training it represents the named entity recognition rules. While early studies were mostly based on handcrafted rules, recent ones use supervised machine learning techniques as a way to automatically induce their rule base. Other named entity extractors are extending the rule induction to semi-supervised or unsupervised learning methods.

Supervised Rule Learning
The current dominant approach for addressing the named entity extraction problem is the supervised rule learning. These approaches base on a tagged test corpus, whose words are manually annotated as positive or negative examples. Extracting rules from a manually tagged corpus refers then back to classical machine learning algorithms, such as *Hidden Markov Model*, *Decision Trees*, *Maximum Entropy Models* or *Support Vector Machines* (SVM).

Semi-Supervised Rule Learning
Semi-supervised rule learning enhances the latter supervised method with the idea that only "bootstrapping" is necessary to start the learning process.
A set of seed rules or names is manually provided to the system. Starting from this knowledge base the system automatically adapts to new training patterns by inducing rules. As an illustration, a semi-supervised method could be initialized with a manually selected list of terms. It then searches for sentences that contain one of these words and tries to learn rules from the sentences' structure. Two (intermediate-) results can be produced in this process. First and most important, rules for the final classifier that point

to named entities and second an enhanced list of seed words –learned by means of the new rules. Semi-supervised methods allow to guide the training process in each step by discarding wrong rules and seed words. If this human-correction step is omitted just one misleading rule can drive these systems to converge to wrong results (e.g. underfitting or overfitting).
Examples of these rule learners are i.e. Collins and Singer's part-of-speech pattern learning [CS99] or Brin's regular expression based approach [Bri98].
Collins and Singer identify rules as re-occurring patterns kept in a spelling-context relation. As an illustration, a proper name followed by a noun-phrase in apposition would retrieve "Maury Cooper" in "Maury Cooper, a vice president at S&P". Starting from initial seed spelling rules, such as "If spelling is *New York*, then it is a *Location*" or "If spelling contains *Mr.*, then it is a *Person*", the corpus is investigated to find new context rules and to find further spelling examples.
Brin uses lexical features in order to generate a list of book-author pairs. Starting from seed examples, such as *[Isaac Asimov, The Robots of Dawn]*, he automatically generalizes the title to a regular expression. As one example, he could find from "The Robots of Dawn, by Isaac Asimov (Paperback)" other book titles and author relations, such as "The Ants, by Bernard Werber (Paperback)". A more detailed selection and description of semi-supervised rule learning systems can be found in [NTM06].

Unsupervised Rule Learning
Unsupervised rule learners are operating on a untagged corpus without positive/negative training examples. They usually use clustering techniques, context similarity measures or word-sense disambiguation related techniques.
For example, Evans [Eva03] uses the World Wide Web to find hypernyms of capitalized word sequences. For instance, when X is a capitalized sequence, the query *such as X* returns nouns that can be chosen as a hypernym to X.
Etzioni et al. [ECD+05] uses Pointwise Mutual Information - Information Retrieval, such as described in Section 3.2.3, as a feature to assess that a named entity can be classified under a given type. For each entity noun-phrase candidate (e.g. *London*) they retrieve a large number of automatically generated discriminator phrases, like *is a city*, or *capital of* etc. These discriminator phrases can later be used as lexical rules for the real task: finding named entities.

3.5.3 Event Extraction

Reading news takes great effort, because tracking the main events throughout the news article is only possible when the user reads the entire story. Although topic detection and tracking techniques have been developed to promptly identify and keep track of similar events in a topic and monitor their progress, the cognitive load remains for a reader to

digest these reports.
In order expedite the research in the field of event-detection, the *Defense Advanced Research Projects Agency* (DARPA) sponsored from 1998 to 2004 the *Topic Detection and Tracking* (TDT) research initiative. Goal of this project and the associated follow-up projects *(Automated Content Extraction Program (ACE)* and *Message Understanding Conference (MUC))* was to investigate the state-of-the-art in finding and following new events in a stream of broadcast news stories. The TDT problem consists of three major tasks: (1) segmenting a stream of data, especially recognized speech, into distinct stories; (2) identifying those news stories that are the first to discuss a new event occurring in the news; and (3) given a small number of sample news stories about an event, finding all following stories in the stream [ACD+98]. More specifically, the TDT research initiative defines the *topic retrospection* subproblem. It incorporates all solutions that identify a set of events in an existing corpus of related stories. Events are defined by their association with stories, and therefore the goal is to group the stories in the TDT study corpus into clusters. Each cluster should represent an event and every news story in the cluster should discuss the corresponding event [ACD+98].
Event extraction approaches can be subgrouped into three approaches: Pattern-based-, event-oriented-, and sentence-based techniques. A selection of the most important event extraction mechanisms will be given in the following under this subcategorization.

Pattern-based Approaches
Within *Message Understanding Conferences* (MUC) one task is called the *Scenario Template* (ST) task which puts the focus to "extract prespecified event information and relate the event information to particular organization, person, or artifact entities involved in the event" [MP98]. Hence, so called *pattern-based* event extraction approaches use information extracted from a text to fill in appropriate fields in predefined event templates. As the task depicts the templates are domain-specific and known a priori.
The most salient representatives of the pattern-based approaches are *Snowball/Stat-Snowball* ([AGP+01]; [ZNL+09]), KnowItAll [ECD+04] and Textrunner [BCS+07]. All of these systems have the common feature to rely on a subject-verb-object extraction for their knowledge base construction. Starting from the precomputed information, the systems differ in their approaches for pattern-matching, feature-based machine learning and natural language parsing.

Event-oriented Approaches
Event-oriented approaches follow the intuition that many of the occurring events will be reported more than once, thus leading to different event descriptions, both within the same document and within topically related documents. Accordingly, event-oriented systems take advantage of these alternate descriptions to improve the event extraction consistency.

One incremental approach stems from Ji and Grishman [JG08]. It uses document-wide and cluster-wide frequency statistics to infer *triggers* and *trigger arguments* that are associated with the particular event types. Subsequently, the obtained classifiers are improved, corrected, or removed in accordance to the (local and cluster-wide) inference rule confidence.
Naughton et al. [NSC08] take a different approach. They focus on merging descriptions of news events from multiple sources. In order to do that their identify the spans of text corresponding to the various events that it mentions. Then, they cluster event descriptions from different articles, so that they all refer to the same event. As a baseline for clustering process they use the agglomerative hierarchical clustering (as described in Section 3.1.1), which they extend by a text sentence position feature. The procedure concludes with a conversion of the event description into a structured form.
A further clustering approach is from Filatova and Hatzivassiloglou [Fil04]. They apply textual clustering methods to group similar paragraphs into topically-related clusters based on primitive or composite features. In order to reduce complexity they ignore sentences that do not contain at least two named entities or frequent nouns. To form the summary for an event one paragraph per cluster is selected and presented to the user.
In order to enhance the granularity of clustering approaches, such as Filatova and Hatzivassiloglou, Liu et al. [LLWL07] defined *Event Term Graph*, which is a semantic network of verbs assessed through a Stronger-than relation. Figure 3.14 (left part) shows this semantic relation graph for a sample test set. On the event term graph, the researchers apply a *DBScan* clustering algorithm, in which the two primary parameters *Eps* denotes the searching radius from each term and *MinPts* corresponds to the minimum neighbors of the term. In addition to the verbs, Liu uses action nouns to approximate the event terms which characterize the event occurrence.

Sentence-based Approaches
Recently, a new *Semantic Role Labeling* (SRL) approach was presented by Yaman et al. in [YHTT09] for the event extraction task. It aims to derive detailed semantic information from a sentence by using a specialized syntactic parser. In comparison to full semantic parsing, SRL only labels semantic roles of constituents that have a direct relationship with the predicates (verbs) in a sentence. SRL's semantic roles include *agent*, *patient*, *source*, *goal*, etc., which are connected to a *predicate*. Optionally, descriptors, such as *location*, *time*, *manner*, and *cause* can be derived. Yet still, while this approach proves to deliver good results, its computational complexity does not allow its application on a larger corpus.

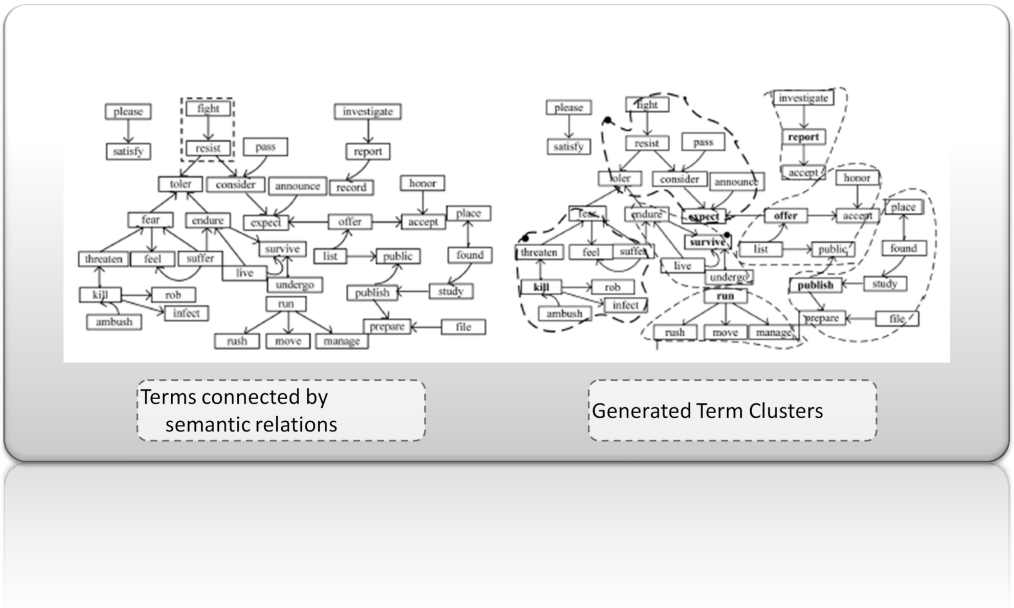


Figure 3.14: Liu's Term Clustering Approach for Event Extraction [LLWL07].

3.6 Text Data Visualization Techniques

The interactive process of analyzing large amount of news has to be built upon a human-computer interface that allows the exploration of patterns within the news cluster. Most of the news exploration systems show only rudimentary visualizations and leave the user with the most important choices: Which news cluster is important with respect to the development in time? Or which news article/news event stands out of the mass?
Text data visualization techniques help in this case to guide the user through the process of identifying important news from the news cluster.

3.6.1 NewsMap

NewsMap is a private user-driven project by Marcos Weskamp [New11]. Its primary layout algorithm bases on a treemap visualization technique that helps to display the steadily incoming news on a large screen. Treemaps algorithms resize rectangles according to their importance and arrange them under the screen sized space-constraints. The author notes on his project page [Mar11] that Newsmap's objective is to provide a tool to divide information into quickly recognizable bands which, when presented together, reveal underlying patterns in news reporting across cultures and within news segments in constant change around the globe. Figure 3.15 shows the main screen.

NewsMap uses the Google News aggregator, which automatically groups news stories with similar content. The size of each cell in the visualization is determined by the

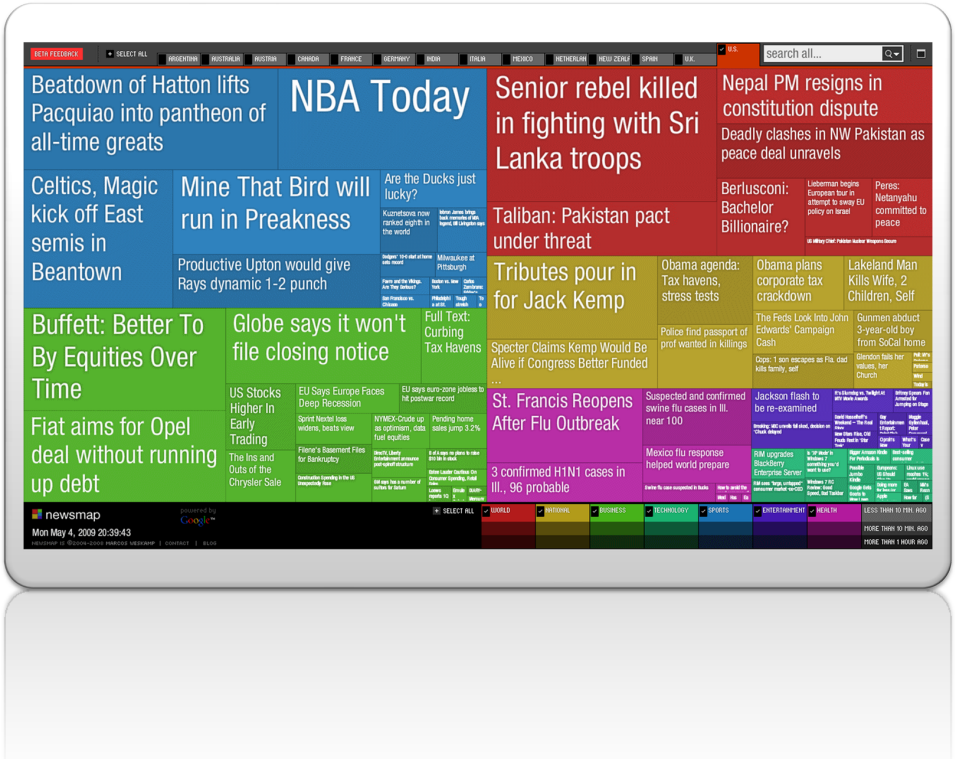


Figure 3.15: NewsMap Treemap Visualization [New11].

amount of related articles that exist inside each news cluster. The users can therefore quickly identify which news stories have been given the most coverage, viewing the map by region, topic or time. Through this process NewsMap still accentuates the importance of a given article.
Newsmap also allows to compare the news landscape among several countries, making it possible to differentiate which countries give more coverage to –for example– more national news than international or sports rather than business (adapted from [Mar11]).

3.6.2 ThemeRiver

One of the most interesting approaches for visualizing text over time is *ThemeRiver*. Havre et al. present in [HHN00] a technique that visually presents the strength changes of individual keywords in a text collection over time. Therefore, ThemeRiver uses the metaphor of a river flowing along a time axis. The broader the river gets, the more important is its corresponding keyword to that time.

In order to assess a keyword's importance the ThemeRiver approach maintains a top-X keyword list for each time step and calculates each keyword's volatility change from time step to time step. If the keyword gains importance its width is increased. If

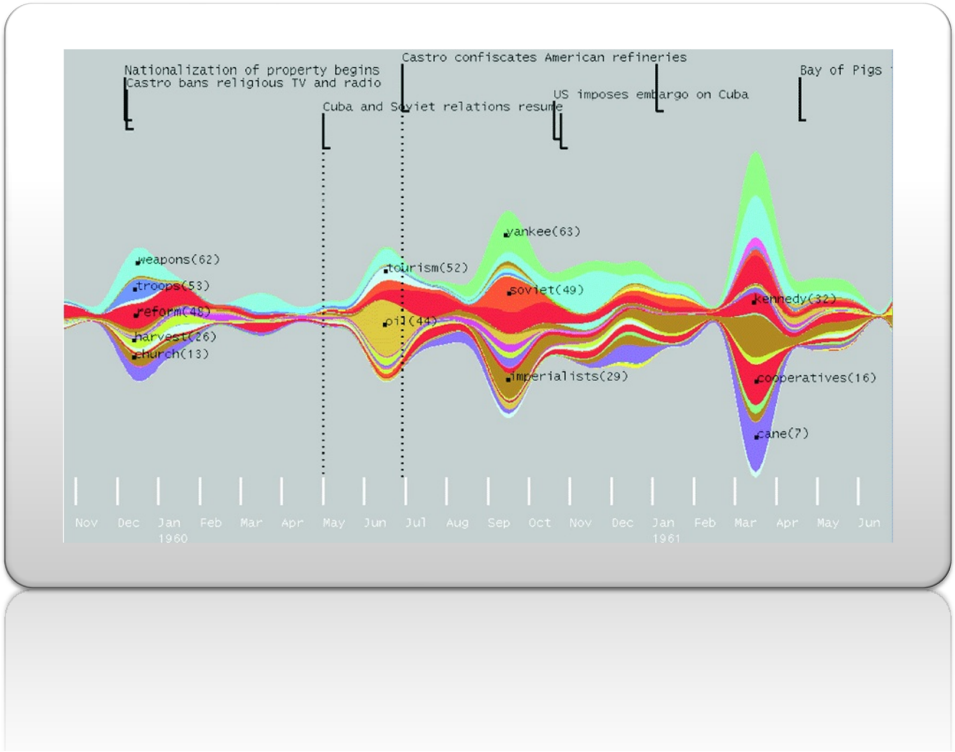


Figure 3.16: ThemeRiver Visualization [HHN00].

its ranking declines it width will be reduced accordingly.

3.6.3 NewsRiver and LensRiver

In [GLYR07] Ghoniem et al. present two time-series related broadcast news video visualizations *NewsRiver* and *LensRiver*, which are inspired by the ThemeRiver approach.

The two interactive video broadcast visualizations allow an exploratory analysis of large scale broadcast video collections. In NewsRiver, a river metaphor is used to depict the thematic changes of the news over time. The interactive lens metaphor in LensRiver allows the playback of fine-grained video segments selected through the river overview. In order to facilitate a multi-resolutional navigation a hierarchical time structure as well as a hierarchical theme structure is presented to the user. Hence, themes can be explored hierarchically according to their thematic structure, or in an unstructured fashion using various ranking criteria. A rich set of interactions such as filtering, drill-down/roll-up navigation, history animation, and keyword based search are also provided [GLYR07]. Most importantly, the two visualizations give means to get an overview about a large collection of news while still allowing the user to assess the content on a much deeper level. Besides the fact that NewsRiver and LensRiver are targeted for video broadcast

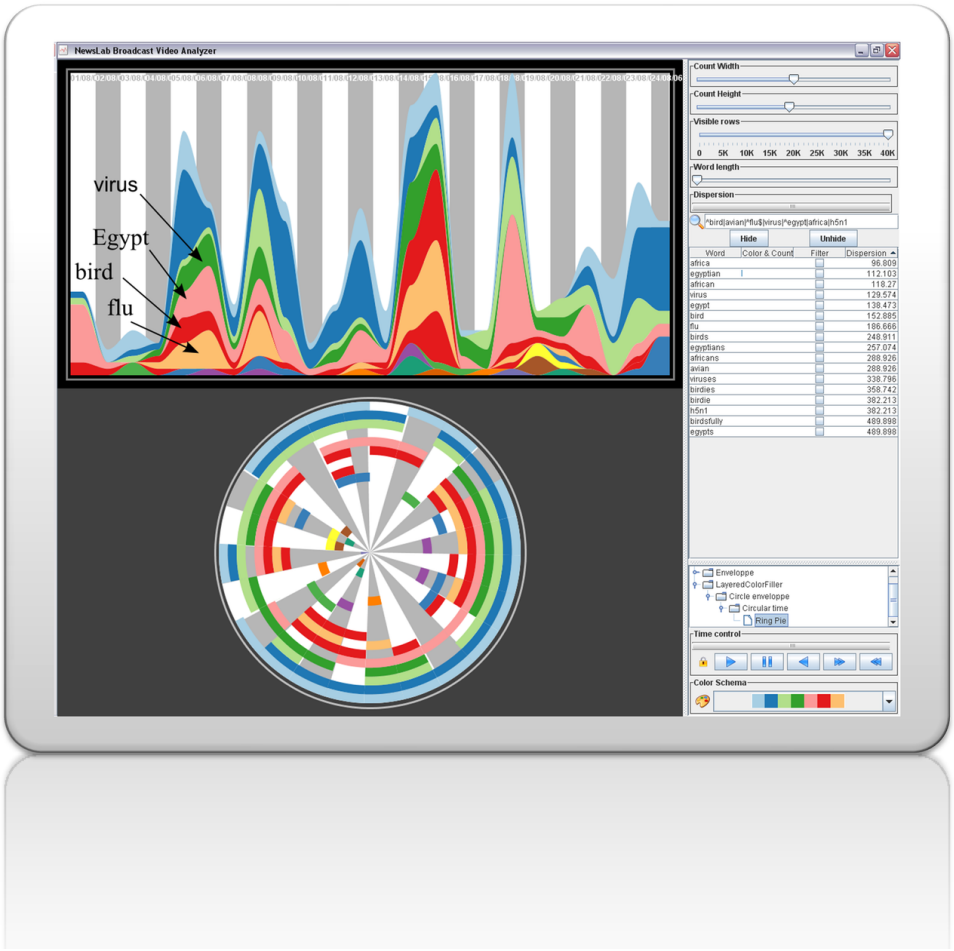


Figure 3.17: NewsRiver/LensRiver Visualization [GLYR07].

events, its basic visualization techniques were inspired by *ThemeRiver* (see: Section 3.6.2).

3.6.4 Event River

In 2009, Luo et al. presented another ThemeRiver related approach, called *EventRiver* [LYK+10].

EventRiver visualizes real life events with temporal references that are extracted from a multi-topic news corpus. This corpus is processed in order to extract document clusters, which can be mapped subsequently to real life events. The width and height of each event river bubble is related to the news clusters semantics and its temporal influences. These characteristics are referred to the continuous attention the news clusters drew over time and stands in relation to other news clusters, or news topics.

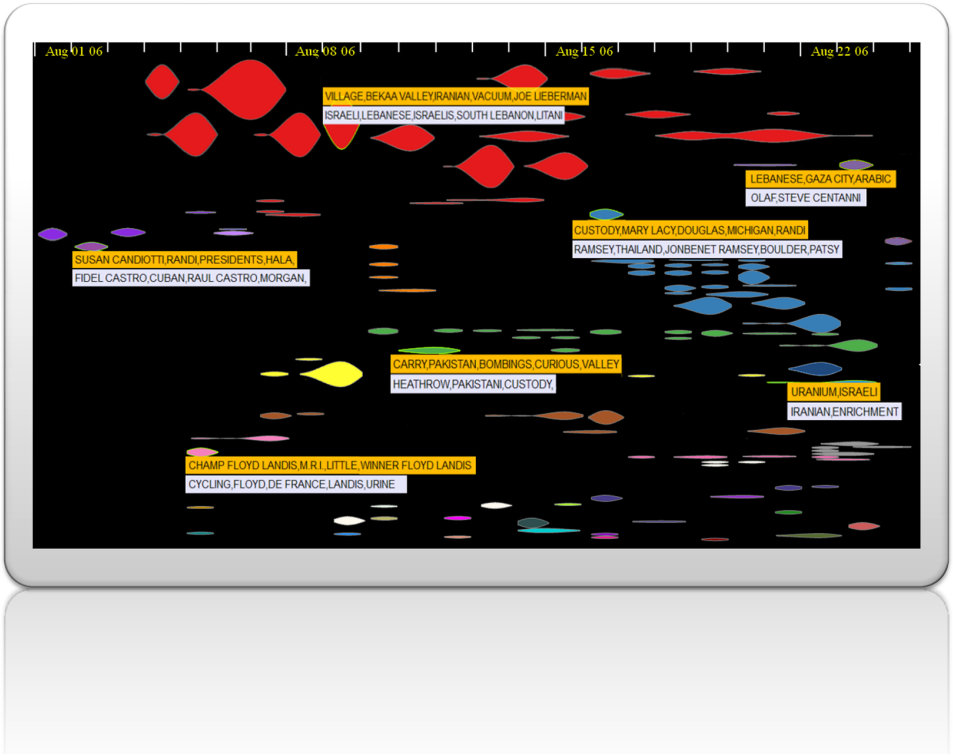


Figure 3.18: EventRiver Visualization [LYK+10].

The user can interactively explore a large collection of news without reading any of the news' full-texts. The main use-cases for the EventRiver visualization are event browsing, event search, tracking, and association, and event investigation. From the overview visualization, which is depicted in Figure 3.18, the user can detect the major events along with their semantics, temporal context and attention without reading one of the documents. Furthermore, the implementation allows users to search events by keywords or example text, to track the evolution of an event of interest, and to examine the possible relationships among multiple events within the temporal context. On the lowest detail level users can examine the event-related documents of interest in full detail and conduct investigative analysis (adopted from [LYK+10]).

3.6.5 Parallel Tag Clouds

Collins et al. present in [CVW09] a visualization that differs significantly from the metaphor of a flowing news/event/topic river. Their approach, called *Parallel Tag Clouds*, is shown in Figure 3.19.

The parallel tag clouds visualization combines graphical elements from parallel coordinates, originally described by Inselberg in [Ins85], and traditional tag clouds. The

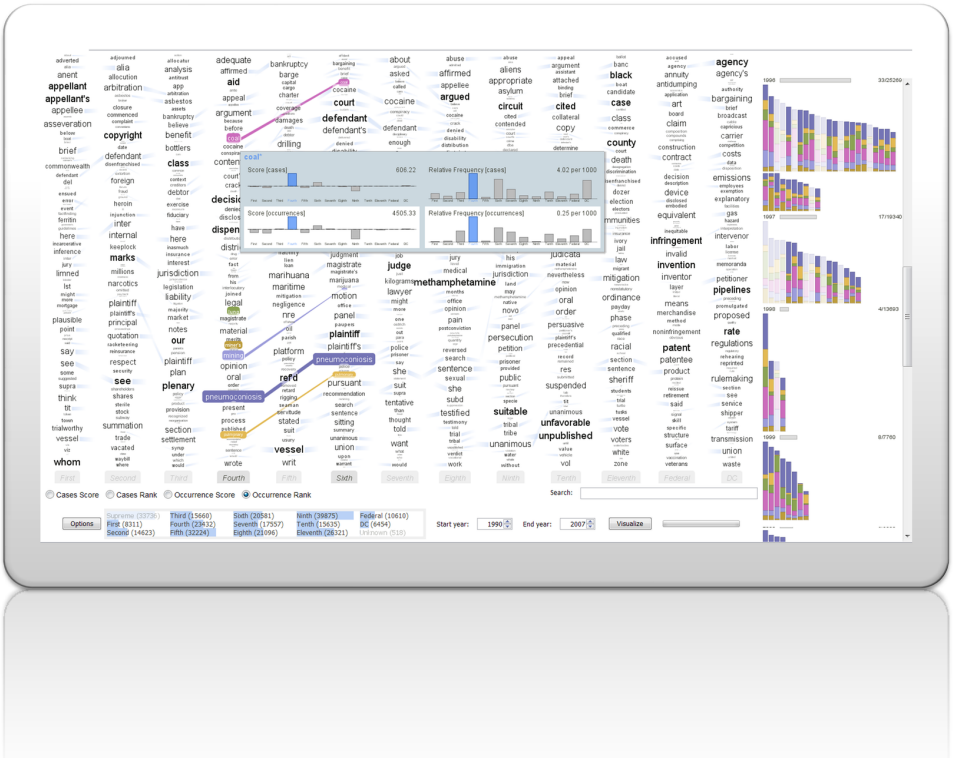


Figure 3.19: Parallel Tag Clouds Visualization [CVW09].

technique offers an overview of a document collections over time. Each of the columns presents the extracted keywords and the keyword's size can be mapped to the relative importance for the specific time interval. In order to present a keyword's development of importance the visualization links two subsequently occurring items while preserving its sizes. In addition to the main features, a detail-view reveals enhanced statistics, such as the global/corpus score, corpus term-frequency, as well as time-interval dependent score and occurrences.

Chapter 4

NewsGuide – News Data Exploration System

This chapter presents *NewsGuide*, a general purpose framework for text data exploration with the primary focus to assess and understand interrelations between news from multiple sources, covering one specific news topic. In accordance with this focus and the main news-topic related questions, presented in Section 1.2 "Primary Objectives", the system helps to guide users in the exploratory analysis of news data.
Due to the fact that the central questions for the news exploration are quite diverse (see: Section 1.2.2 "Project Objectives") and required significantly different approaches, we chose to investigate structural- and semantic aspects separately. Figure 4.1 shows an overview of NewsGuide's processing pipeline. It starts with the raw news text data. Due to a partnership between EMM and the University of Konstanz, the meta-data enhanced news stream is available for our research department. Despite the fact that most meta-data is valuable, such as the named entity extraction or the geo-location extraction, NewsGuide's focus leaves little to no use for it. Hence, exclusively the news' full-text, which had to be retrieved manually from the contained URL, is used for its processing- and visualization steps. Consequently, NewsGuide is loosely coupled with the EMM processing pipeline, while its exploration functionality is coherent in itself.
After the news selection step, which is primarily the fetching of a time-interval subset of the entire news stream, a preprocessing and filtering module builds the basis for an enhanced feature extraction. Textual-, structural- and semantic features are extracted by their specific modules and are presented in their corresponding visualization (overview, structural view and semantic view). Each of these subbranches has a different intention.

- Structural View: The structural view enables the user to explore inter-document text coherency. Its primary aim was to develop a visualization that allows exploratory and confirmatory analysis on the data. Here, time-interrelations and textual similarity measures help to extract non-obvious patterns.

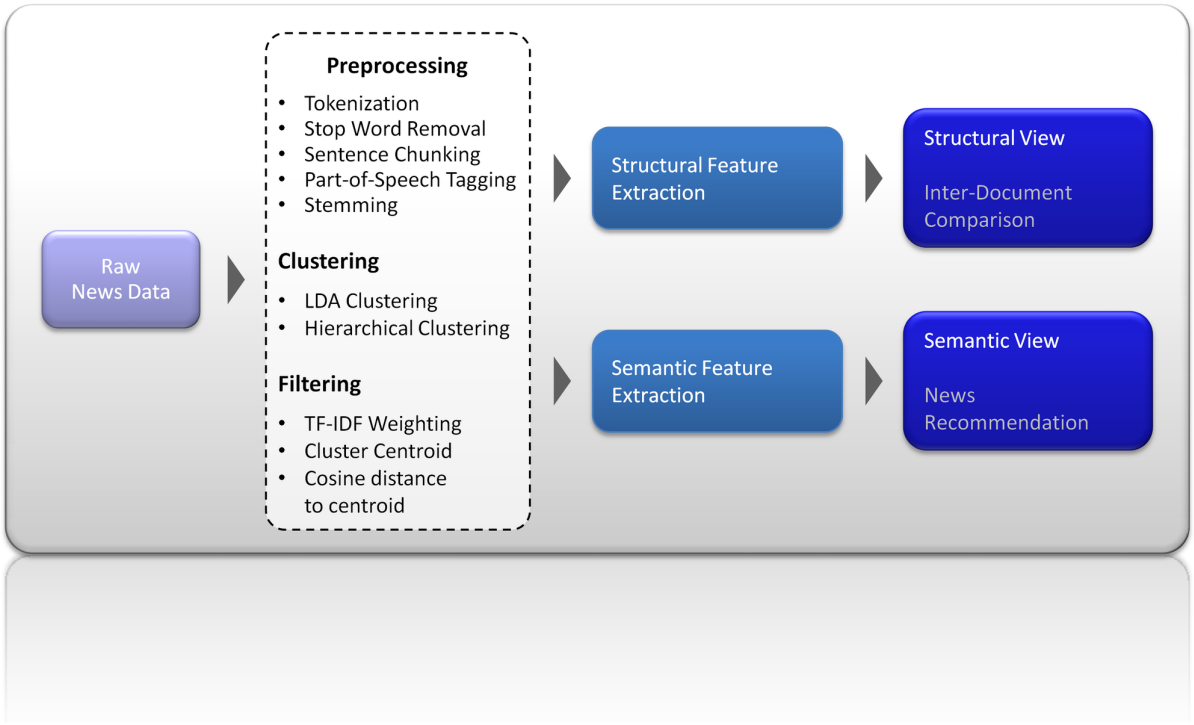


Figure 4.1: NewsGuide Processing Pipeline.

- Semantic View: The semantic view stands out from the latter mentioned component, since the visualization stands not in the primary focus, but rather the engineering techniques guiding the user through an interactive reading process. It has to be seen as a news recommendation approach that enhances the users' reading experience and provides new solutions for the mentioned problems in Section 1.2 "Primary Objectives".

In the following, each important technical and visual design decision will be described in detail, discussed and compared to the main approaches in this field.

4.1 Data Preprocessing

Data preprocessing lays the foundation on which the most important steps are based. It functions, on the one hand, as a necessary step for data cleaning (i.e. in the case of stop word removal) and on the other hand is inevitable for the subsequent chained processing stages, such as extractive summarization or main storyline extraction. A filtering mechanism is also integrated into NewsGuide. This is due to the fact that without such a step the sheer amount of data would be neither processable nor visualizable. Moreover,

it would overwhelm the user in the visual analytics tasks. The following section will give an insight into the used data preprocessing techniques.

Tokenization

Tokenization is the process of splitting a full-text into individual tokens, typically words. Due to the ease of implementation NewsGuide system uses the *LingPipe* natural language processing toolkit. LingPipe (accessible at [Lin11a]) offers free and paid licenses. The free version stands under the *Royalty Free License*. It allows users to "copy or modify the Software or use any output of the Software (i) for internal non-production trial, testing and evaluation of the Software, or (ii) in connection with any product or service you provide to third parties for free" [Lin11b].
In the NewsGuide implementation, the tokenizer is customized to a pipeline (see: Builder Design Pattern, i.e. described in [GHJV95]) that uses the standard *Indo-European Tokenizer* as the basis for its work. This general-purpose tokenizer is the most appropriate one for the "standard" English language. Other options are e.g. Medline tokenizers, which are trained on medical test data and incorporate domain-knowledge.
Other implementations were considered and discarded due to the following reasons. For instance, the *Apache OpenNLP* tokenizer proved to perform slower than the LingPipe implementation. Another option was to implement a natural language tokenizer myself. This approach was rejected, due to the many special cases that have to the taken into concern. In addition to that highly-sophisticated natural language processing methods can parse the grammatical structure of the text to pick significant terms or chunks (sequences of words), such as noun phrases. One example for a grammatical parser/tokenizer is presented in [MB04]. One dependency parsers implementation is the *Standford Parser* –described in [MMM06]. It was discarded, since it is sufficient for the NewsGuide implementation to calculate only standard word tokens.

Stop Word Removal

The process of removing stop words, such as "and", "or", "the", etc., incorporates a comparison of every word in the article to a compilation of known words in the stop word dictionary. Thus, *stop word removal* is a dimensionality reduction technique with the aim to remove terms, which are not thought to convey any meaning as a dimension in the vector space.
NewsGuide takes advantage of LingPipe's integration of this step into the tokenization engine. While the tokenization takes place, a stop word removal implementation filters the token set. In order to enhance LingPipe's insufficient standard stop word dictionary, a significantly enlarged stop word list was chosen. This list is available under [Tex11a].
A different approach is implemented for the semantic feature extraction, which will be described in Section 4.4. It uses the output of a part-of-speech –short POS– tagger to reject all tokens with a specific POS tag. In NewsGuide's semantic feature extraction

implementation this technique is used to reject prepositions, coordinating and subordinating conjunctions, and proper names. These POS tags are removed, because they are either are not available in the used WordNet implementation or do not incorporate any meaning to the sentence at all.

Sentence Chunking

One important preprocessing step for the structural feature extraction is the *sentence chunking*. The sentence chunker decides to split sentences according to a list of possible stops (i.e. ".", ")", etc.), impossible penultimates (i.e. personal and professional titles, ranks, address abbreviations, etc.) and impossible starts (i.e. ",", ";", "%", etc.).
Sentence chunkers are trained classifiers segmenting text into its constituent sentences. NewsGuide uses a Indo-European sentence model. This heuristic sentence model was retrieved from a combination of several corpora. Many sentence chunker implementations exist. Many of them use a domain-dependent sentence model, such as LingPipe's MedLine sentence model, which has been trained on 2.000 biomedical texts.

Part-of-Speech Tagging

Part-of-Speech (POS) tags are an important and valuable feature when assessing a word's meaning. For example, nouns and verbs incorporate usually more meaning than adjectives and adverbs. Tagging tokens with their specific POS requires a probability model that helps to assign the correct tag out of the predefined tag set.
A large number of POS taggers are available for the research community. The most important ones are the *Stanford POS tagger*, the *TreeTagger*, *LingPipe's POS tagger*, *Apache NLP POS Tagger*, or *MorphAdorner*.
In order to evaluate these POS taggers two primary evaluation criteria have to be regarded. The most important one is the precision rate. According to the POS accuracy evaluation from Matthew Wilkens in [Wil08a], the precision must be evaluated relative to the importance of the POS type. Large errors on rare tags generally matter less than modest errors on common tags. Overall, MorphAdorner performs best on the reference data and LingPipe's implementation is marginally better than Stanford's tagging approach or the TreeTagger.
The second evaluation criterion is speed. The implementations vary to a great extent in their processing speed. A performance evaluation of the mentioned POS taggers can be found in [Wil08b]. The results depict that the TreeTagger outperforms the other mentioned taggers significantly. Figure 4.2 shows the results in a bar chart visualization.

In the diagram, LP denotes LingPipe, SL stands for Stanford left3words model, SB is Stanford's bidirectional model TT is the TreeTagger, and MA is the MorphAdorner tagger implementation.
In accordance to these evaluation results and fact that a dependency to the LingPipe library exists anyways, NewsGuide used the LingPipe POS tagger implementation for

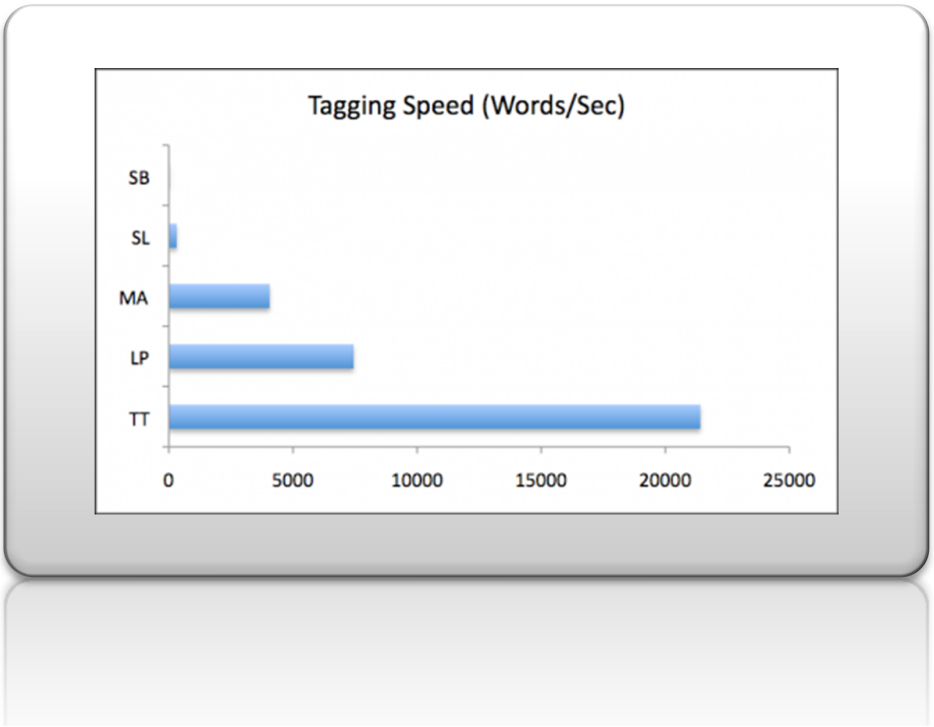


Figure 4.2: Part-of-Speech Tagger Comparison: Speed [Wil08b].

this preprocessing step.

Clustering

The clustering of news data has the goal to find coherent news article groupings. Each of these clusters should be assigned to one specific topic. The property of coherence denotes in this context that all articles, assigned to the cluster, should have the same topic and all articles, which are not assigned to it should have different topics. In other words, the clustering algorithm's goal is to find a partitioning of the article set that fulfills some objective criterion based on the similarity measure and the text representation.

One important aspect is that the project's problem definition defines the underlying data model to be static and all visualizations do not have to incorporate means for handling a "streaming data model". Consequently, the data could be assumed to be available. Therefore, the focus of this project was not to evaluate or develop sophisticated text clustering methods, but was rather to investigate a given topical-coherent textual data model/cluster. Nonetheless, it was still necessary to develop a baseline data set on which the text-, structural- and semantic feature extraction mechanisms could be tested.

As already noted in the introduction (see: Section 1.2) the news data originates from the EMM news stream. On a daily bases over 40.000 news in 43 languages arrive on this stream. Accordingly, it is inevitable to choose a subset. This subset was chosen to be all English news in the time span of *October 2010 [2010-10-1]* to *December 2010 [2010-12-31]*. In the end, 455297 news were extracted for the text data clustering.

As described in Section 3.1, clustering techniques for text data have two promising directions: online/stream/incremental clustering and offline/static text data clustering. Due to the reason that the goal of NewsGuide is to help the user to develop a better understanding of a static topic-coherent news cluster, a probabilistic clustering method was chosen and implemented in the form of the Latent Dirichlet Allocation (LDA) (see Section 3.1.2 for a detailed description).
In all LDA implementations various parameters have to be selected empirically or with the help of approximation methods. For example, NewsGuide's LDA implementation uses a sampling technique, called *Gibbs Sampling*, for an estimation of the LDA language model parameters. The iterative Gibbs sampling simulates a high-dimensional distribution by sampling a lower-dimensional subset of variables, where each subset is conditionally dependent on the latter ones. This can be seen as a special case of the widely known *Metropolis* algorithm, described for example in [Has70]. The LDA topic estimator produces samples consisting of a topic assignment for every token in the corpus. The initial/first sample is produced by randomly assignment from topics to tokens. Then the sampler works iteratively through the corpus, one token at a time. At each token, it samples a new configuration given all the topic assignments to other tokens in the corpus. The sampler is instantiated with the assumed number of topics (here: 50 topics are assumed), the minimum probability for assigning topics to documents (topic prior; chosen to be 0.1), and a word prior estimation, which controls the minimum likelihood for a word-to-topic connection (chosen to be 0.01). These estimates were chosen in accordance to hints in [Wan07], [BNJ03] and [YMM09].
All in all, the Gibbs sampling for LDA proves to be parameter sensitive. Its main disadvantage is that the number of clusters has to be known a priori, as well as the topic-document prior. If the both priors are not appropriately chosen they tend to lead to a skewed posterior distribution and ultimately an incoherent clustering.

From the initially large set of selected news articles, LDA extracted a series clusters. From these clusters –also corresponding to one topic– five have a satisfying quality and are interesting (primarily due to their topic salience during the selected time period). The five selected topics are described by LDA with the keywords depicted in Table 4.1.

The columns *Min Distance*, *Max Distance* and *AVG Distance* depict one way to measure the cluster's coherence. It is calculated as the minimum, maximum and average inter-document cosine distance. The standard deviation column denotes the articles'

Topic ID	Description Keywords	Number of Documents	MIN Distance	MAX Distance	AVG Distance	Standard Deviation
Topic 10	world cup win football game top league lead final beat	36710	0.0	1.0	0.04133	0.36332
Topic 15	israel israeli talks peace gaza palestinian east us palestinians state	33902	0.0	1.0	0.04567	0.3728
Topic 22	killed pakistan afghanistan afghan kills attack nato kill suicide us	34392	0.0	1.0	0.4013	0.3819
Topic 30	assange wikileaks probe against bail pm govt scam case founder	33604	0.0	1.0	0.0399	0.3536
Topic 44	eu ireland irish crisis euro debt bailout budget imf plan	34890	0.0	1.0	0.0421	0.3412

Table 4.1: Latent Dirichlet Allocation: Extracted Clusters with Keywords and Coherence Measures

average standard deviation from the artificial cluster centroid. The centroid was created from the top 100 Tf-Idf keywords taken from all documents.
It should be mentioned that the clustering of the over 450.000 articles took several days to run (on a notebook computer with 4Gb Ram and an Intel dual-core CPU as of December 2010).
This run time is directly proportional to the number of topics times the number of tokens in the corpus and results in the fact that LDA –on the one hand– is highly scalable/parallelizable, but –on the other hand– is fairly slow per epoch.

Filtering
Displaying the huge amounts articles per topic (as for example shown in Table 4.1) in an understandable fashion is nearly impossible and reveals its own problems (i.e. overplotting, high-dimensionality, etc.). Yet still, it should be noted that some rare approaches, such as *Pixel Bar Charts* [KHDH02], allow to display such huge amount of data points. In order to reduce the amount of articles to a meaningful size and to increase the coherency of the topic clusters NewsGuide filters the data to retrieve only the most important documents. Its filtering mechanism works as follows:

Algorithm 4.1: Filtering Mechanism

```
1.  Filter all stop words.
2.  Calculate the TF-IDF scores for every remaining
    token in the cluster.
3.  Build an artificial cluster centroid with the top x tokens.
4.  Calculate the cosine distance between every document and
    the centroid.
5a. Either reject all documents above a predefined threshold
5b. Or rank all documents according to their centroid cosine distance
    and reject all documents above a ranking index.
```

With the help of Algorithm 4.1 the clusters were reduced significantly to 50-60 articles per cluster.

4.2 Structural News Assessment

Structural features are one of the main pillars in the NewsGuide news data exploration system. With these features questions regarding plagiarism (copying of text), text modifications and inter-document similarity patterns can be answered. The main goal of the structural comparison is to lead the user to non-obvious patterns in the topic-coherent news cluster. Examples for outstanding insights could be that one news agency regularly adapts/copies parts of another news agency's text or that an earlier news article text influences other later published news.
Moreover, the goal definition incorporated that it should be possible to compare news articles visually on a structural layer. Thus, the structural feature visualizations should enable the user to investigate if a consistent, but not mandatory article structuring is used in the online news sphere.
The primary structural features extracted from the news articles are the article length (stop-word-removed token length versus its overall text length), number of paragraphs, number of sentences, and so on. In the subsequent step, these features can be compared by means of five implemented similarity measures, several filtering- and sorting options. The five similarity measures range from simple and well-understood text overlap similarity measures, such as the *Jaccard-* and *Cosine* similarity, as described in Section 3.2, and incorporate also semantical similarity measures, such as the *Semantic DTW Distance*, *Google Distance* and a *Bag-of-Synset* related *WordNet BoS Vector Similarity* measure, which will be described in detail in Section 4.3.

4.2.1 Structural View - Inter-Document Comparison

The structural view was developed as the main component in project phase 1. Its visualization bases on a matrix of document thumbnails that allows an overview-to-

detail exploration. The project goal definition leads to the Figure 1.6. In NewsGuide's structural comparison component the user should be guided visually to the interesting inter-document comparisons. As the "Project Phase 1 Objectives" diagram in Figure 1.6 depicts, the overview bases on the user-selected inter-document distance scores and allows after that a more detailed comparison on the structural level and lastly on the textual level.
In accordance to these goals, the interface in Figure 4.3 was developed.

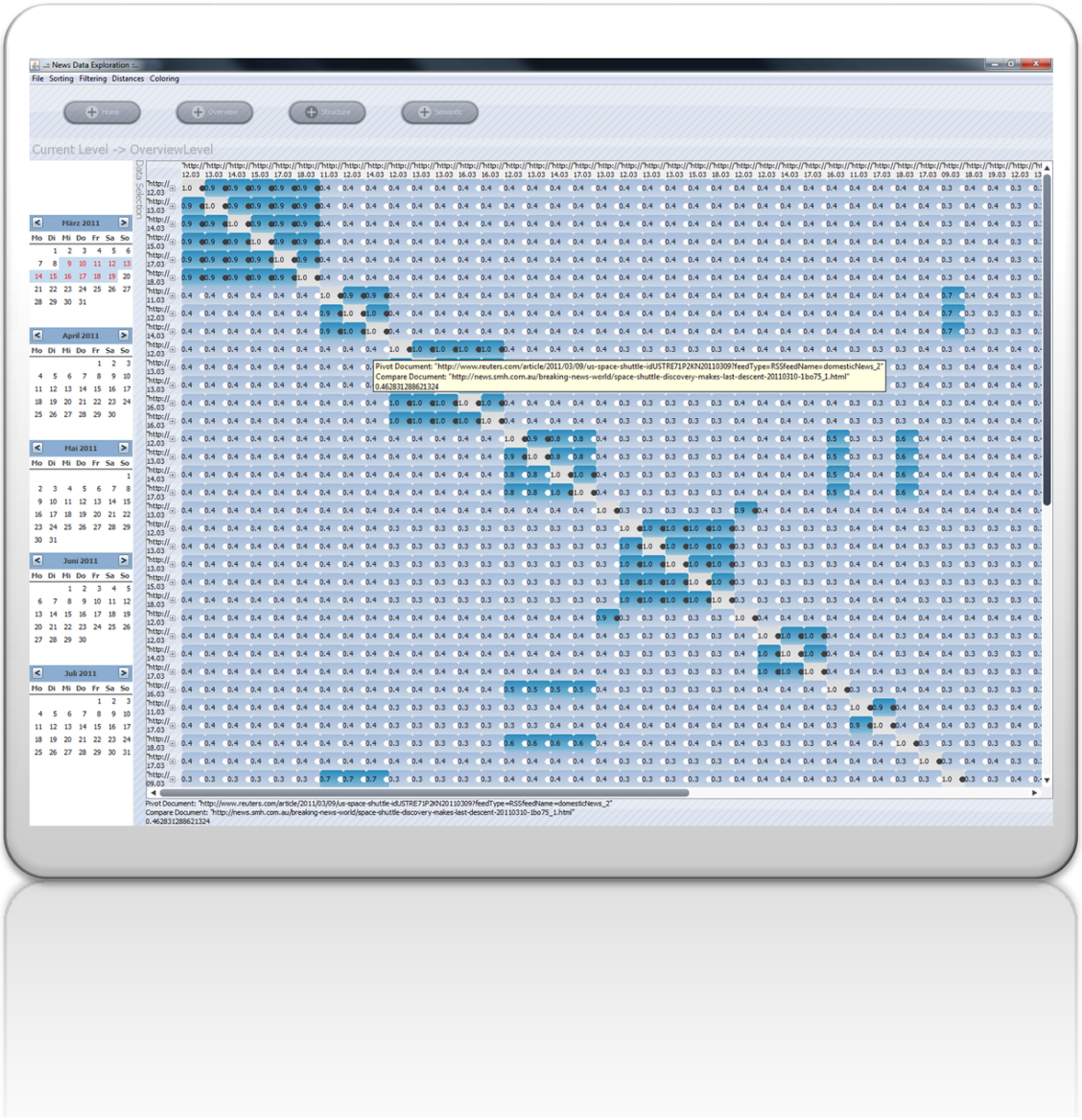


Figure 4.3: Structural View - Overview.

The Figure 4.3 depicts the main visual interface of the structural inter-document comparison view. It is build up from three information/filtering subcomponents and the main view.
On the left side, a time-interval chooser allows to filter news in a specific time range. All articles, which do not fall into the selected time range will be neglected, thus potentially leading to a sparse matrix view.
Most interaction possibilities are given the user through the menu bar on top of the screen. Here, the user can –for example– select *sorting options*, such as different or same sources first (for an inter-/intra- news agency comparison), time-descending (for a temporal analysis of the news development) or ascending/descending distance aggregate sorting options.
In Figure 4.3, the view is sorted according to a descending distance aggregate, promoting the documents with the highest inter-document comparison aggregate to the front. Moreover, the menu bar enables the user to *filter* all recognized updates. The next option, *Distances*, allows to choose from one of the six distance/similarity measures (Jaccard, Cosine, Semantic DTW, Google, and BoS distance). In the Figure 4.3, the cosine distance is showcased. Lastly, the user can choose from several (discretized or continuous) heatmap *color codings*. The discretized color codings differ in their amount of discretization steps and the base color. In the Figure 4.3, a purple-to-blue color coding with three classes –extracted from [Bre11]– is visualized.
In addition to that a small information panel on the bottom shows context-dependent information, such as the article id, numerical scores or meta-data.

The main panel consists of a matrix view in which each cell corresponds to the comparison of the pivot document (always correspondent to the row ID) to the comparison document (correspondent to the header ID). The demonstrated numerical score is computed by means of the chosen textual similarity measure and color coded with the selected or default option. A logarithmic color-to-distance mapping is implemented to emphasize more important distance intervals. Furthermore, each cell contains a small black/white dot. It gives a visual clue whether the articles stem from the same source (black dot) or not (white dot).
After having a numerically supported overview on the data, the user can choose to expand one or more rows in order to explore the structural features among each other. Figure 4.4 a zoom onto this primary interaction feature. Here, *document thumbnails* are rendered. They encode visually the sentence- and paragraph structure, as well as their textual similarity score (always in comparison to the row-header pivot document). Moreover, the coloring stays consistent to the overview, thus justifying the overall scores (i.e. if the document has 50% strongly similar sentences and 50% without a similarity, an overall similarity score of 50% would be presented).

In the Figure 4.4, one can see a good example for the introduced *influencing/copying pattern*. The view is configured with the sorting option "Time Ascending", "No

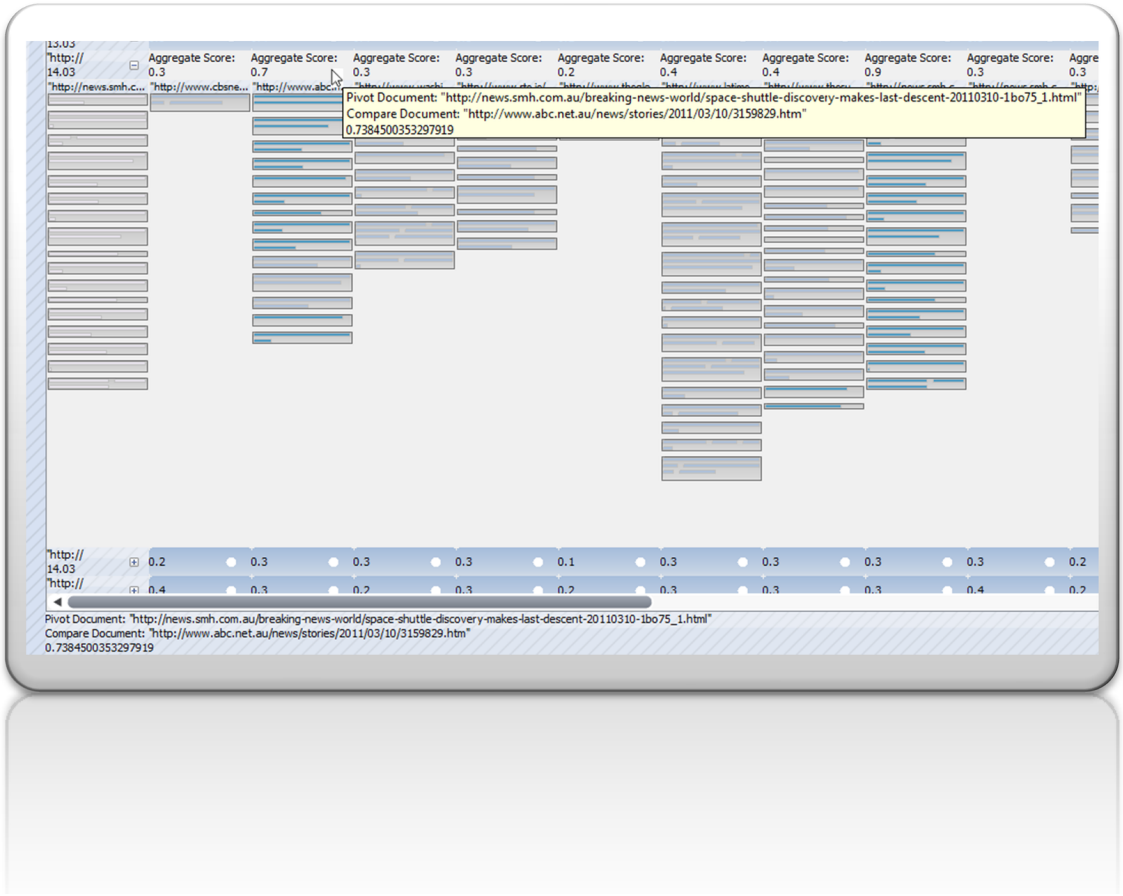


Figure 4.4: Structural View - Expanded Row.

Updates", and the "Cosine Distance". Every document with a high similarity score –depicted by a dark blue color– and a later publishing date –can be derived from the row- and column header (here only row header is visible due to presentation purposes)– is suspect to copying. These parameter apply for example for the second column after the pivot element row header on the left side of the Figure. The copying hypothesis is even more obvious if it is not the same source that wrote the article (the information panel or tool tip text show source1 "news.smh.com.au" and source2 "www.abc.net.au"). In the further investigation one can see that a large amount of sentences –here also the paragraphs– are in the highest similarity class, depicted by the darkest blue color. Only four of 14 sentences are not rated "very similar".
For further investigation, the user has to switch from the structural view to the textual representation. This component is shown in Figure 4.5. The text detail view shows the pivot and comparison text in the left, respectively right text panel. Besides comparing the text by reading both articles, the user is supported by the consistent color coding

and the corresponding range slider on the bottom of the screen. With the range slider, the user can decide to highlight all sentences within a high-similarity interval (i.e. above 80%) or see the minimum and maximum similarity values.
Proving the hypothesis that the comparison text was copied from the original source is easy here, because the right panel reveals that four sentences with 100% agreement exist. Furthermore, four sentences have neglectable changes in the range of 80% to 99% and two sentences were only partly copied (50% to 80% agreement). Four sentences seem to be written by the author (under 50% agreement).
Additionally, the user can choose a sentence and its related sentence and compare it visually with the help of the *Diff* algorithm, which visually marks insertions/deletions (Figure omitted).

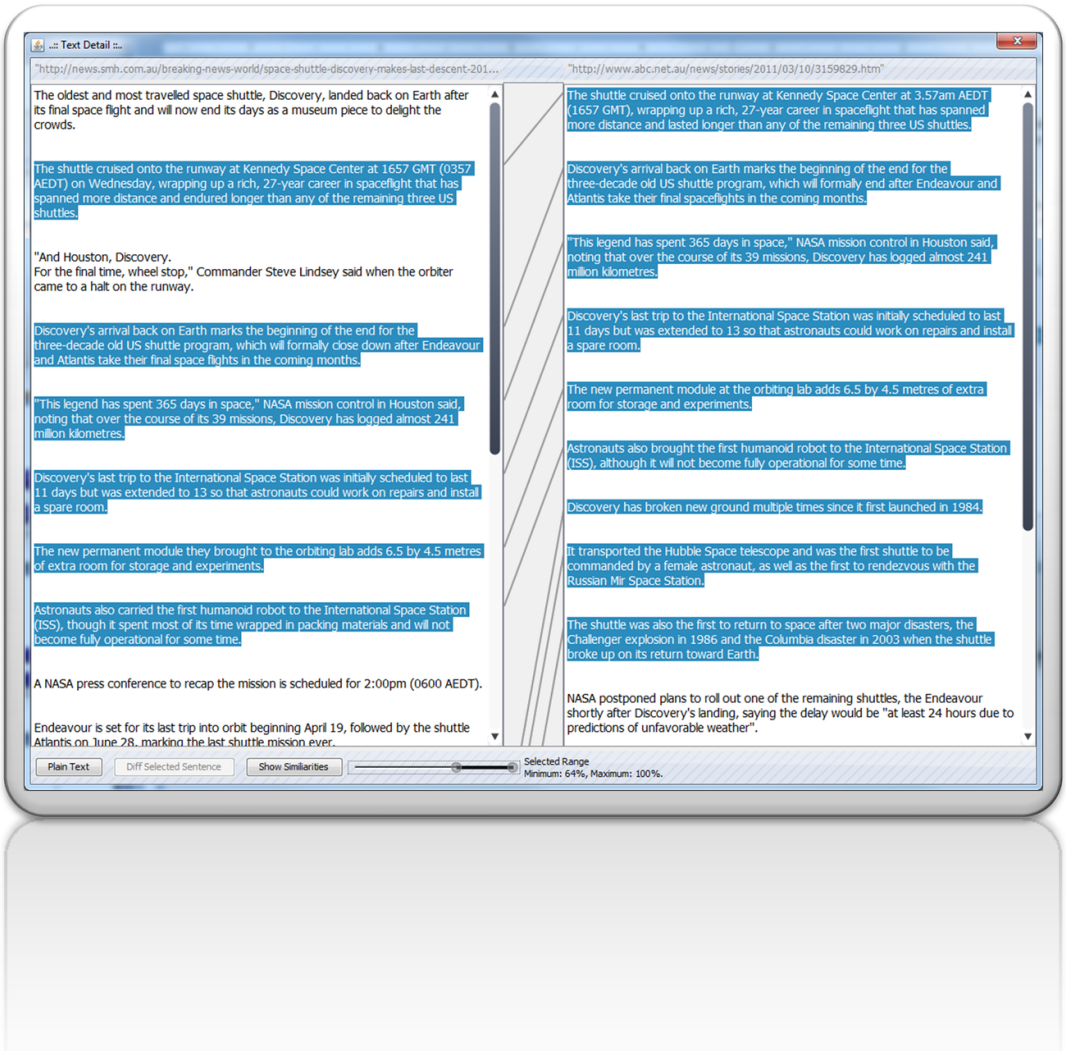


Figure 4.5: Structural View - Text Detail View.

4.3 Bag-of-Synsets Model and its Applications

The *Bag-of-Synsets* (BoS) model is an extension of the *Bag-of-Words* (BoW) model, which has originally been suggested by Degemmis et al. in 2006 [LZL06]. Its fundamental idea is to model the meanings (senses) corresponding a word, rather than the word itself. The main building block of BoS is a so called *Synset*, which represents the natural language's polysemy. In a synset, or *synonym set*, a word is related to all terms with a very similar, identical or synonymous meaning. Thus, every word in a synset can be seen as a placeholder that could potentially replace the original word without losing a sentence's meaning.
For a topical-related semantic investigation of the similarity between words, sentences, paragraphs and ultimately articles, the BoS model can be seen as a key-enabler. BoS builds a semantic layer on top of the usually employed Bag-of-Words model with its strict correspondence to one word in the vocabulary. With a synonym-driven approach, such as the BoS model, one can expect to find topically/semantically-related phrases, which could not been found with the classical BoW model. As a direct consequence several application areas can benefit from this semantic layer, as Figure 4.6 depicts.

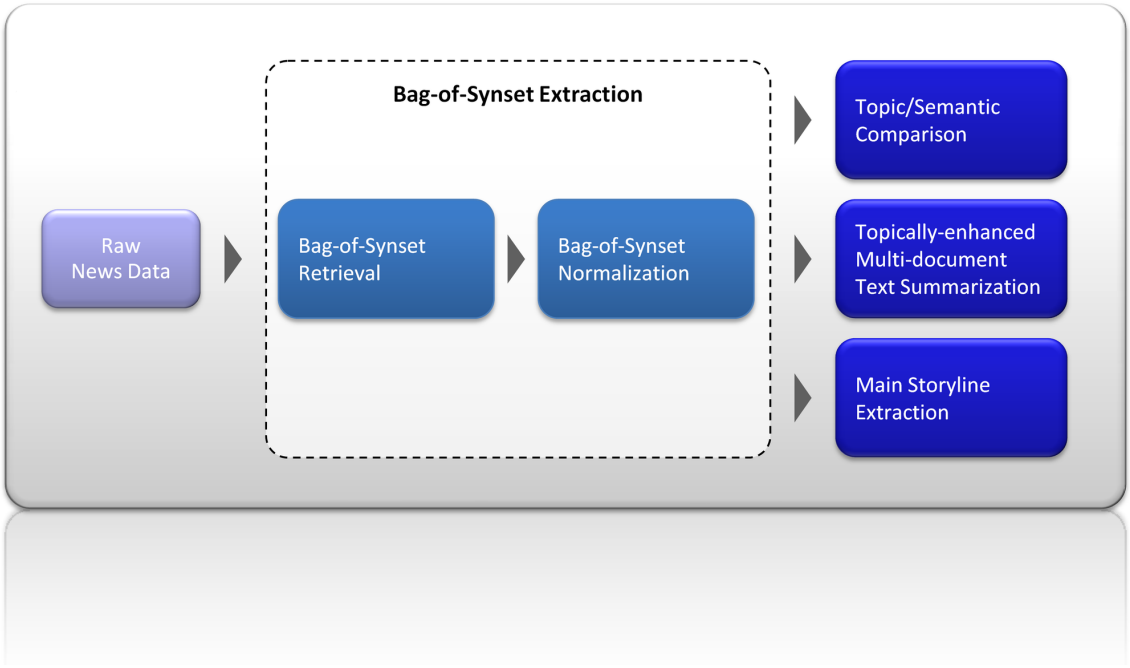


Figure 4.6: Bag-of-Synsets Model and its Applications.

As Figure 4.6 depicts the Bag-of-Synset model allows to enhance several text mining research fields. One example for its application is the topical-relatedness comparison,

which will be presented in Section 4.4.1. Here the goal is not to find documents that reveal a high word overlap, but rather to retrieve articles, which talk about the same topic or subtopic with potentially different wordings. Another application area is multi-document text summarization (explained in Section 4.4.2), which uses the BoS similary measure to find salient topically-related sentences that form a summary. Lastly, a new subfield –the main storyline extraction– will be presented in Section 4.4.3. The main storyline extraction tries to develop statistical means for analyzing and retrieve an article's primary storyline.
All of the mentioned BoS application areas base on an extraction mechanism, which will be presented in the following.

4.3.1 Bag-of-Synsets Retrieval

Retrieving an article's Bag-of-Synset model requires inevitably a lexical knowledge database, such as WordNet [MBF+90], which has an automatically or human-guided approach to assign words to their polysemic senses.
The procedure for extracting an article's BoS model is the following:

Algorithm 4.2: Bag-of-Synset Retrieval Algorithm

```
1.  For each word w in the article.
2.  Skip w if it is a stop-word.
3.  Query the WordNet implementation for all synsets of w.
3a. If no value is returned assign w to
    a static ``unknown'' synset.
3b. If multiple synsets are returned choose the most likely.
4.  Store the synset ID in the article's BoS list.
```

The algorithm 4.2 has three outstanding steps, which will be described in the following:
First, step #3 issues a query to the lexical database WordNet. This query is not as obvious at it looks, since it incorporates word stemming and a lexical analysis step. The word stemming is one of the particularly noteworthy preprocessing steps. Standard morphological analysis methods, such as the suffix stripping *Porter Stemmer* [Por97] operate in a rule-based manner reducing words to a potentially artificial base form. As one example, the Porter stemmer reduces "cats" to "cat" (with the rule "$s \mapsto \emptyset$ "). In this simple example the plural word is reduced to the existing singular word. However, in often occurring cases artificial words are created. For example, "caresses" will be reduced to "caress" (with the rule "$sses \mapsto ss$") or "ponies" will be reduced to "poni" (with the rule "$ies \mapsto i$"). While this has implementation and efficiency advantages and is appropriate for the major use-cases, it has the outstanding disadvantage that

artificial words cannot be found in a lexical database, such as WordNet. In order to find a better solution to this problem, NewsGuide develops a new stemming procedure. It stems the words entirely on a WordNet basis. Given a word, it first retrieves the correspondent part-of-speech (POS) tag and then queries the noun-, verb-, adjective- or adverb database for the manually-tagged word base form. This guarantees that a stemmed word can definitely be retrieved in the database. Note that, the POS tagger operates on a probability distribution. Thus, it can happen that the wrong POS tag is assigned to a word. In this case a word's base form will be searched in all other databases, except the one that failed before.

The second outstanding line is step #3a (assigning to an "unknown" synset). News articles in nearly all cases contain special terms, which cannot be found in the predefined WordNet database. Thus, it becomes necessary to construct a synset that acts as a collecting vessel. The "unknown" synset contains particularly all proper nouns, author-"invented" words, colloquial- and slang terms. This implementation works fine in the NewsGuide case. However, it shall be noted that this limitation mainly stems from the dependency to WordNet. If all synonym sets were created manually, it could be possible to retrieve all words-to-synset relations. In this case it would be necessary find the existing umbrella or hyponym terms, which subsume special cases (i.e. "an iPhone is a smartphone is a handset is a telephone"). As one approach to this problem one could –for example– use a Pointwise-Mutual-Information approach –such as described in Section 3.2.3– to extract synsets from a large corpus. This would have the advantage that every possible word-synset correlation could be found. Yet still, this theoretical approach suffers from its computational complexity and thus its infeasibility.

The third outstanding line in Algorithm 4.2 is step #3b, which can be subsumed under the term of *Word Sense Disambiguation* (WSD). Word sense disambiguation is a (young) research field on its own. The goal is to develop statistical techniques for assigning the best possible match in the word-to-senses relation. Most common is the subdivision into four approaches: Dictionary- and knowledge-based methods, supervised-, semi-supervised- and unsupervised methods.
Dictionary and knowledge-based methods are built on the hypothesis that word-sense relations can be observed from predefined *dictionary definitions*. Hence, whenever a dictionary description, matching the query word, can be found then the word's definition functions as a hint for the word's senses. One example for a dictionary and knowledge-based method is the *Lesk algorithm* [Les86]. It tries "to decide automatically which sense of a word is intended (in written English) by using machine readable dictionaries, and looking for words in the sense definitions that overlap words in the definition of nearby words" [Les86].
Supervised methods are based on the assumption that a word's context –i.e. all words before and after the word– can provide enough evidence on its own to disambiguate

words. Accordingly, this approach makes world knowledge and reasoning sources unnecessary. Supervised methods embody the classical classification problem, which can be solved by *Support Vector Machines*, *Decision Trees*, *probabilistic classifiers (HMM)* and other classifiers. These techniques have to work on a potentially very high-dimensional feature vector and can thus be slow and feature-dependent.
Semi-supervised methods follow the idea of bootstrapping as described in Section 3.5.2. Here, a small amount of seed data functions as an indicator for rules which can be extracted from a manually-tagged corpus. Subsequently, the learning algorithm tries to find new and confident classification rules which can be used for extracting further relations.
Unsupervised methods, also called *Word Sense Induction* methods, operate on a fully untagged corpus. Their assumption stems from the fact that similar senses occur in similar contexts. Accordingly, senses could be induced by means of grouping word co-occurrences. One exemplary approach in this field is described by Schütze in [Sch98].

A much closer investigation for the word sense disambiguation research field can be found in [Nav09] or [ZH05].
Due to the ease of implementation, NewsGuide's approach to word sense disambiguation is fairly straight-forward. In accordance to the WordNet Api it uses the most common synset for a word. When querying the WordNet 3.1 database it returns an ordered list of synsets. This ordering is dependent on the importance a word has for a synset (rated by humans). Consequently, it allows for a low-level approximation of the WSD problem. On the other hand, it is not as accurate as the other approaches mentioned beforehand and gives room for future improvements.

4.3.2 Bag-of-Synsets Normalization

The next important step towards a fully functional BoS text representation is normalization. BoS normalization is a required step without which a further investigation would not be possible. The BoS normalization outcome is a length-normalized BoS vector, where the component i in document D_x can be compared to the i^{th} component in document D_y and both components point to the same synset. Algorithm 4.3 shows the normalization approach.

Algorithm 4.3: Bag-of-Synset Normalization Algorithm

1. Retrieve all synsets in a set implementation $AllSynsets$.
2. Assign a unique index from the range $[1,\ldots,|AllSynsets|]$ to each synset s.
3. For each document d construct a BoS representation with the length $|AllSynsets|$.
4. Retrieve the synset list d_s of document d

5. For each synset in the synset list d_s increment a counter at the precomputed index position.
6. Return an unmodifiable BoS representation for d

This straight-forward algorithm leads to a length-normalized BoS representation of all documents in the cluster and allows in the following step a semantic comparison. In addition to that, it has one very positive side effect as Table 4.2 shows. One can see that the BoW representation reduces the dimensionality of the representation vector between 45% and 52% dependent on the unique token count. This has a clear impact on the computation performance, because most of the measures' calculation times grow proportionally to the dimensionality.

Topic ID	Bag-of-Words Vector Size	Bag-of-Synset Vector Size	Dimension Reduction
Topic 10	4981	2268	0.4553
Topic 15	3022	1469	0.4861
Topic 22	2379	1097	0.4611
Topic 30	5349	2456	0.4592
Topic 44	3493	1666	0.4760
Discovery Topic	1537	810	0.5270

Table 4.2: BoS's Dimensionality Reduction Impact

4.4 Semantic News Assessment with the Bag-of-Synsets Model

Semantic features are one of the cornerstones of the NewsGuide exploration system. They enable the user not only to see a structural and textual coherence, but also the topical relatedness. One technique to assess the semantics of a text is the BoS text representation, which abstracts away from the words while still preserving the sentences' semantics. A further means is the usage of semantic relatedness measures on the text itself. Approaches, such as the *Google Similarity Measure*, *Resnik's Context Similarity Measure* or the *PMI-IR Approach* of Turney (all described in Section 3.2.3), also discard the textual word representation and go over to a conceptual view onto the words.

4.4.1 Article Comparison

The proposed BoS model abstracts away from the words themselves and leads to a more topic-related view onto the data. Therefore, it allows the user to assess topical similarity. One approach to measure this relatedness is derived from the idea of correlation-coefficients. The more common synsets are shared between the documents, the more topically and semantically related are those two documents.
Having a length-/component-normalized representation, such as the ones calculated in the normalization step (see: Section 4.3.2), a similarity measure can be set up. One similarity that incorporates the latter mentioned idea is the following:

$$S_{18}(D_x, D_y) = \frac{|s(D_x) \cap s(D_y)|}{|s(D_x) \cup s(D_y)|} \tag{4.1}$$

where $s(D)$ retrieves the normalized BoS representation of a specific document D, $s(D_x) \cap s(D_y)$ lists the overlapping synset entries and $s(D_x) \cup s(D_y)$ constructs an artificial vector in which a dimension component is set if it exists in one of the vectors D_x or D_y. Alike the cosine coefficient the distance measure depicted in Equation (4.1) holds the theoretical case in which one document would compared to the concatenation of another document with itself, the BoS distance results in the same value as without the concatenation ($S_{18}(D_1, (D_2 \circ D_2)) = S_{18}(D_1, D_2)$). Hence, not the amount of synset occurrences in one document is of interest, rather than its overall co-occurrence across the documents.

4.4.2 Topic Summarization

One application that shows the power of semantics in the news data exploration is *summarization*. Summarization, more specifically extractive summarization, selects parts of the text –usually sentences– and presents an overview/excerpt that incorporates the text's primary information. The main techniques and approaches are already depicted in the Section 3.3 "Text Summarization". Yet still, while this research field reaches back into the 1990s and had most publications in the end of the 1990s, it still lacks a good evaluation method. One reason for this is that even user-generated summaries differ significantly and noticeably in the prioritization of facts. In order to tackle this problem most papers bring their own small test set. It is often derived from the *Document Understanding Conferences* (DUC) [DUC11] and modified (i.e. translated or filtered). In accordance to that the LexRank summarization algorithm (described in Section 3.3.3), also comes up its own eleven sentence test set. It was extracted from one DUC 2004 cluster (*DUC2004; d1003t cluster*). Table 4.3 depicts LexRank's test sentences.

Sentence No.	ID	Text
1	d1s1	Iraqi Vice President Taha Yassin Ramadan announced today, Sunday, that Iraq refuses to back down from its decision to stop cooperating with disarmament inspectors before its demands are met.
2	d2s1	Iraqi Vice president Taha Yassin Ramadan announced today, Thursday, that Iraq rejects cooperating with the United Nations except on the issue of lifting the blockade imposed upon it since the year 1990.
3	d2s2	Ramadan told reporters in Baghdad that Iraq cannot deal positively with whoever represents the Security Council unless there was a clear stance on the issue of lifting the blockade off of it.
4	d2s3	Baghdad had decided late last October to completely cease cooperating with the inspectors of the United Nations Special Commission (UNSCOM), in charge of disarming Iraq's weapons, and whose work became very limited since the fifth of August, and announced it will not resume its cooperation with the Commission even if it were subjected to a military operation.
5	d3s1	The Russian Foreign Minister, Igor Ivanov, warned today, Wednesday against using force against Iraq, which will destroy, according to him, seven years of difficult diplomatic work and will complicate the regional situation in the area.
6	d3s2	Ivanov contended that carrying out air strikes against Iraq, who refuses to cooperate with the United Nations inspectors, will end the tremendous work achieved by the international group during the past seven years and will complicate the situation in the region.
7	d3s3	Nevertheless, Ivanov stressed that Baghdad must resume working with the Special Commission in charge of disarming the Iraqi weapons of mass destruction (UNSCOM).
8	d4s1	The Special Representative of the United Nations Secretary-General in Baghdad, Prakash Shah, announced today, Wednesday, after meeting with the Iraqi Deputy Prime Minister Tariq Aziz, that Iraq refuses to back down from its decision to cut off cooperation with the disarmament inspectors.
9	d5s1	British Prime Minister Tony Blair said today, Sunday, that the crisis between the international community and Iraq did not end and that Britain is still ready, prepared, and able to strike Iraq.
10	d5s2	In a gathering with the press held at the Prime Minister's office, Blair contended that the crisis with Iraq will not end until Iraq has absolutely and unconditionally respected its commitments towards the United Nations.
11	d5s3	A spokesman for Tony Blair had indicated that the British Prime Minister gave permission to British Air Force Tornado planes stationed in Kuwait to join the aerial bombardment against Iraq.

Table 4.3: LexRank's Test Set (Extracted from *DUC2004 Cluster d1003t*)

Since NewsGuide provides the user with a summary of the news cluster, these sentences are used to evaluate the summarization approach, too. NewsGuide also takes the direction of a graph-based relevance mechanism, such as in LexRank [ER04]. It is used to find the most salient sentences, which describe the article/cluster in the best possible manner. The reasons for choosing a graph-based relevance mechanism are the following.

First, it is related to one of the most successful information retrieval methodologies, namely *PageRank* [BP98]. The LexRank implementation (described in Section 3.3.3) uses a modified PageRank graph-based relevance algorithm, which leads to a good result accurracy. Second, its methods are extendable and modifiable in comparison to the highly sophisticated and monolithic procedures, described in Section 3.3. And third, it proves to be reasonably good in performance.
The standard LexRank mechanism, as described in [ER04], follows the hypothesis that a sentence is the more salient, the more salient sentences connect to it. Therefore, it builds a fully-connected graph, whose edge-weights are correspondent to the similarity score of the sentence-nodes. In the subsequent pruning-step all edges below a predefined threshold are rejected. Thus, a partially-connected graph is the result. In this graph every sentence is only connected to similar sentences.
Furthermore, LexRank follows the intuition that saliency is related to the inter-sentence comparison score. If a sentence is very similar to all other sentences, then it is potentially important in the entire news cluster. Due to this, the standard LexRank graph-based salience extraction uses the modified cosine distance (shown and explained in Equation 3.55) to extract similar sentences.
With the help of the *Bag-of-Synset* model and its related similarity measure NewsGuide takes this idea one step further. More specifically, it does not measure the textual similarity, but rather the semantical-relatedness of two sentences. This score is used in LexRank's similarity matrix implementation and the associated ranking mechanism. Table 4.4 shows the calculated intra-sentence similarity scores.

	d1s1	d2s1	d2s2	d2s3	d3s1	d3s2	d3s3	d4s1	d5s1	d5s2	d5s3
d1s1	1.000	0.700	0.300	0.182	0.250	0.200	0.200	0.385	0.400	0.200	0.154
d2s1	0.700	1.000	0.500	0.182	0.250	0.200	0.200	0.231	0.300	0.200	0.154
d2s2	0.300	0.500	1.000	0.273	0.167	0.200	0.200	0.154	0.200	0.200	0.154
d2s3	0.182	0.182	0.273	1.000	0.250	0.273	0.455	0.231	0.182	0.182	0.154
d3s1	0.250	0.250	0.167	0.250	1.000	0.500	0.167	0.231	0.333	0.250	0.231
d3s2	0.200	0.200	0.200	0.273	0.500	1.000	0.200	0.077	0.200	0.200	0.231
d3s3	0.200	0.200	0.200	0.455	0.167	0.200	1.000	0.231	0.111	0.111	0.077
d4s1	0.385	0.231	0.154	0.231	0.231	0.077	0.231	1.000	0.231	0.154	0.077
d5s1	0.400	0.300	0.200	0.182	0.333	0.200	0.111	0.231	1.000	0.444	0.308
d5s2	0.200	0.200	0.200	0.182	0.250	0.200	0.111	0.154	0.444	1.000	0.231
d5s3	0.154	0.154	0.154	0.154	0.231	0.231	0.077	0.077	0.308	0.231	1.000

Table 4.4: Bag-of-Synsets Intra-Sentence Similarity Matrix

Taking the similarity matrix from Table 4.4 as a starting point, the graph-based relevance mechanism calculates the importance ranking scores. These results are displayed in Table 4.5. All the values are normalized, so that the each column's largest value is 1. One can see that the sentence *d1s1* has a very high score in every sentence pruning configuration (0.05, 0.1, 0.2, and 0.3). In the first two configurations it is ranked on

the first place and falls back to the second rank in the third and fourth configuration. In contrast to the depicted BoS results, the original LexRank modified cosine distance results are attached in the last two columns for comparison reasons. Here, sentence *d4s1* stands out as the most salient sentence. This result is partially influenced by the fact that this sentence has proportionally more outstanding words than the others. These words (i.e. "representative", "secretary", "cut", "off", "deputy") are –in accordance to the Idf score– promoted and lead ultimately to a higher overall ranking. On the other hand, from the BoS semantic similarity results one can conclude that from a semantic point-of-view sentence *d1s1* is the most salient. It contains the highest number of synsets which match the other sentences' BoS representation. Hence, it has the strongest correlation-coefficient with all other BoS vectors.

	BoS (0,05)	LR	BoS (0,1)	LR	BoS (0,2)	LR	BoS (0,3)	LR	LR (0.1)	LR (0.2)
d1s1	**1,000**		**1,000**		0,913		0,960		0,601	0,694
d2s1	0,986		0,986		0,896		0,836		0,847	0,732
d2s2	0,844		0,844		0,623		0,566		0,349	0,677
d2s3	0,849		0,849		0,760		0,709		0,752	0,655
d3s1	0,916		0,916		**1,000**		0,748		0,591	0,434
d3s2	0,828		0,809		0,612		0,615		0,799	0,872
d3s3	0,746		0,726		0,521		0,709		0,355	0,499
d4s1	0,757		0,717		0,770		0,536		**1,000**	**1,000**
d5s1	0,935		0,936		0,911		**1,000**		0,592	0,740
d5s2	0,800		0,801		0,583		0,581		0,691	0,697
d5s3	0,700		0,661		0,608		0,542		0,592	0,450

Table 4.5: LexRank Salience Scores computed with the BoS Similarity Measure

4.4.3 Main Storyline Extraction

A further interesting application area for semantic relatedness measures is the *main storyline extraction*. Main storyline extraction has the goal to retrieve an article's primary focus or plot from a set of statistical features. These features must allow differentiating between primary and side storylines. If such a main storyline can be found it would be possible to reduce the reading times significantly. A reader could –for example– start his reading process with a storyline-based summary, which highlights only the primary plot. After having decided to continue reading, the user could either choose to reduce the cognitive load by skipping already assessed semantically related information (i.e. paragraphs or sentences) or search for other main storyline-compliant articles to check their point-of-view. In all cases the user would need a visual guiding system pointing him to the already retrieved aspects.
In order to extract a main storyline from a news article, statistical features on the

BoS document representation can be used. It is intuitively understandable, that from an information-theoretic point-of-view the highest co-occurrence of synsets defines outstanding key-synsets. This idea corresponds to the term-importance weighting, which derives a term's saliency from their inter-/and intra-document frequencies (the most important term-weighting mechanisms are described in Section 3.5.1). Combining the descriptions of multiple of these key-synsets can be perceived as a storyline description. The question is therefore how many key-synsets have to be merged in order to find a storyline description. Inspired by the work of Resnik [Res95], [Res93] and Huffman [Huf52] we can define a greedy algorithm that takes advantage of the information-theoretic notion of self-information (e.g. described in [Ros10]). Figure 4.7 shows a diagram of its NewsGuide's main storyline extraction algorithm.

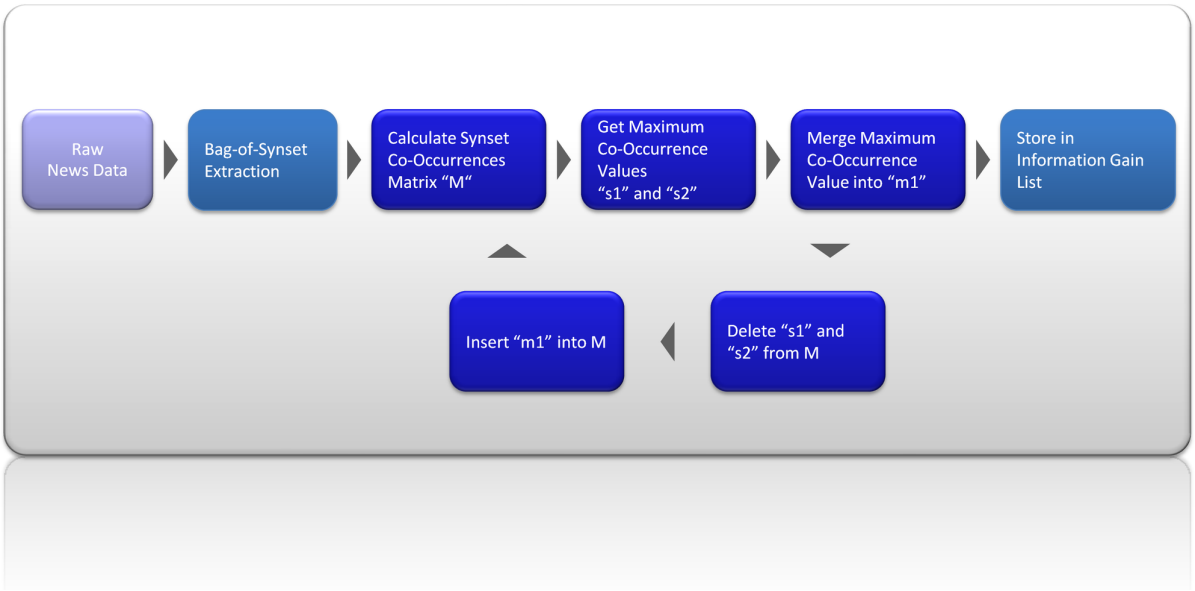


Figure 4.7: Storyline Extraction.

The procedure starts with the BoS extraction for each article in the news cluster. The extraction will retrieve a number of N synsets. Simultaneously, an inverted index is constructed for the mapping of each synset to its article occurrences. This index implicitly stores the topic's synset frequencies.
As the next step, each synset co-occurrence will be calculated. Per definition the synset i and j co-occur, if a document that contains the synset i also incorporates words that belong to the synset j. The co-occurrence count is normalized by the overall cluster size for comparison reasons.
The bottom-up greedy algorithm consists of a main loop, which will be traversed $N-1$

times. Starting from the calculated synset co-occurrence matrix M it extracts the maximum co-occurrence entry and the correspondent synset IDs $s1$ and $s2$. These two synsets are merged into an artificial synset $m1$, which stores –on the one hand– references to the synsets $s1$ and $s2$ and –on the other hand– the co-occurrence value. This co-occurrence value is updated according to the following procedure: Retrieve all articles from the inverted index containing references to synset $s1$ and store them in a list $l1$. Repeat the retrieval with $s2$ and store the articles in the $l2$ list. Lastly, intersect $l1$ and $l2$ to find all articles with co-occurrences to $s1$ and $s2$ and normalize the value.
The resulting merger synset $m1$ will be stored in an information content list, which will be used for the subsequent analysis step. Its purpose is to track the information gain that results from each merge step.
The last two steps will update the co-occurrence matrix M. Therefore, the old references to $s1$ and $s2$ are eliminated (two rows and two columns will be deleted) and the merger synset $m1$ is inserted as their replacement (insertion of one row and column). The last procedural step is the update of the co-occurrence matrix M, which is effectively a recalculation of the last inserted row and column.
In each of the following iterations, the algorithm selects the maximum co-occurring synsets, merges them and updates the matrix M. Note, that not only basic synsets are merged. Rather so, the algorithm allows to combine merger synsets into agglomerative groups consisting of two or more merger synsets. Especially, these groupings are of greatest interest for the user, since their co-occurrence can be seen as a hint for the storyline keywords, respectively -synsets.
Yet still, while this algorithm allows for a reasonable effective extraction of co-occurring synsets, or synset groups, it does not answer the question how many synsets describe the storyline.
At this point the notion of *self-information*, a measure of *information content*, can be taken into account. By definition, the amount of self-information contained in a probabilistic event depends only on the probability of that event. In our example, an event can be mapped to the encountering of a document D in the corpus C that contains words of the synset s. Note, that each value in the co-occurrence matrix M is normalized by the overall cluster size. As an example, if the number of documents in which a synset $s1$ occurs is given by the function $freq(s1)$ and $s1$ can be retrieved in every document, its relative co-occurrence value is $freq(s1)/|C| = 1$. If a synset $s2$ can only be found in one document, then the relative co-occurrence value is $freq(s2)/|C| = 1/|C|$. By definition, this relative frequency score depicts the *probability value* of encountering a document with the synset/synset group in the corpus. Its probability is equivalent to the relative document-frequency of the synset in the corpus. The smaller its probability, the larger the self-information associated with receiving the information that the event indeed occurred. In other words, as probability of an event increases, its informativeness decreases.
With this information-theoretic background, NewsGuide develops a simple, yet effective

mechanism for retrieving the amount of related storyline key-synsets. Remember, the information gain list contains the merging steps in a descending ordering. For each of the merge steps, the synset references and the relative co-occurrence values –or event probability– is stored. According to the standard argumentation of information theory [Ros10], one can compute the information content of the merge events m as the negative the log likelihood, as Equation (4.2) depicts.

$$IC(m) = -\log_2\left(\frac{freq(m)}{|C|}\right) \tag{4.2}$$

The base of the logarithm allows to change the unit of information –for example– from bits ($\log_2(x)$) to nats ($\ln(x)$) or hartleys ($\log_{10}(x)$), which has only an impact on the information-content scale.
Figure 4.8 depicts a line chart of the information content development for one of the news clusters (namely the "Discovery Topic"). It shows on the y-axis the information content for every merge step and on the x-axis the synset merge step index. The blue line shows the information content development.

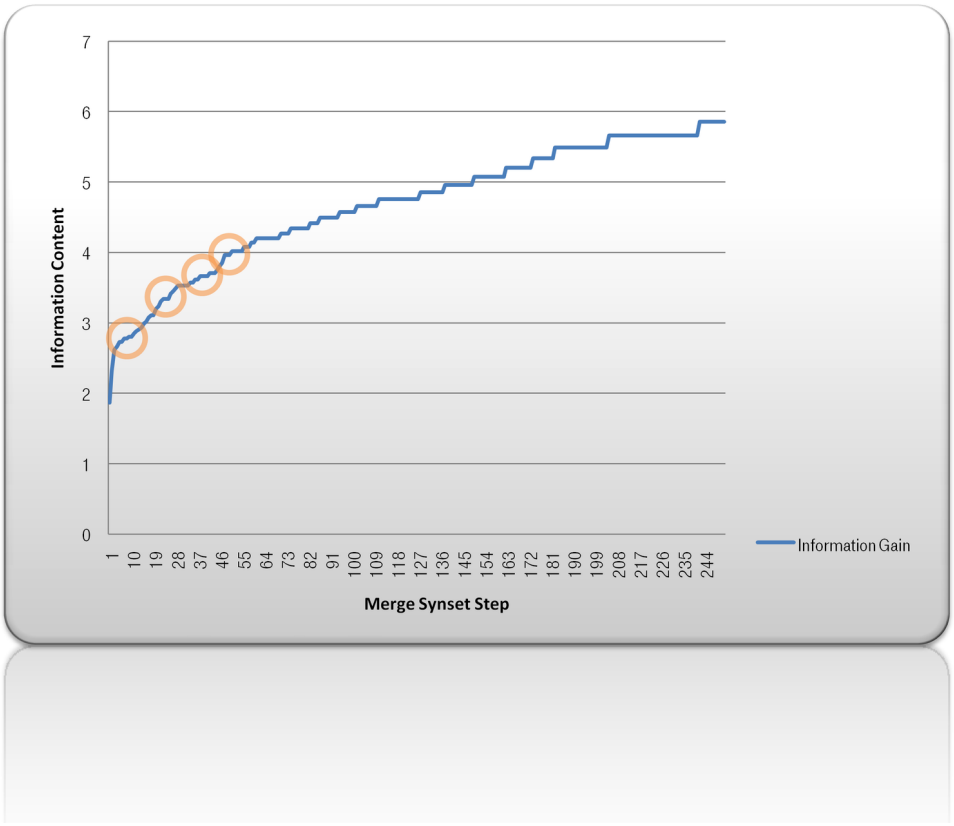


Figure 4.8: Information-Content Plot for the Storyline Extraction.

As Figure 4.8 depicts, the development of the information content is not always steady and reveals plateaus (the first plateaus are highlighted by orange circles). The ends of the plateaus are the most important storyline extraction hints, due to the following reasons. Each end of a plateau refers to point in which the information content significantly changes. This statement corresponds to a series merge events, whose co-occurrence probability is nearly equally likely –thus leading to a plateau of information content. After that the synset merging co-occurrences changes to an unlikely probability, which is reflected by an increasing information content. Hence, a point is reached, which differentiates all following documents from the latter seen document subset. From an article reading perspective, these plateau-endpoints refer to a subset of articles, which all contain a certain amount of keywords. After each endpoint, a new group of keywords is formed, which can be perceived as a new aspect of the story. All merge steps from one plateau-endpoint to the next endpoint therefore refer to a grouping of key-synsets, which are closely related and can be seen as descriptors for the storyline aspects.
In the Figure 4.8 one can retrieve the first information plateau at merge step #9. The corresponding information content list enumerates its synset merge steps:

1. Information Content: 1.8674
   Synset: [**space**, infinite]
   Synset: [**land**, dry_land, **earth**, **ground**, solid_ground, terra_firma]

2. Information Content: 2.3219
   Synset: [**discovery**, **find**, uncovering]
   Synset: [**state**, **nation**, **country**, **land**, commonwealth, res_publica, body_politic]

3. Information Content: 2.6174
   Synset: [shuttlecock, bird, birdie, **shuttle**]
   Synset: [**center**, **centre**, **middle**, **heart**, eye]

4. Information Content: 2.7791
   Synset: [**station**]
   Synset: [**plan**, **program**, **programme**]

5. Information Content: 2.8038
   Synset: [**mission**, missionary_post, missionary_station, foreign_mission]
   Synset: [**concluding**, **final**, **last**, terminal]

6. Information Content: 2.9070
   Synset: [**day**, twenty-four_hours, twenty-four_hour_period, 24-hour_interval, solar_day, mean_solar_day]
   Synset: [**international**]

7. Information Content: 2.9613
   Synset: [**land**]
   Synset: [**astronaut**, **spaceman**, **cosmonaut**]

8. Information Content: 2.9894
   Synset: [**orbit**, celestial_orbit]
   Synset: [**scope**, **range**, **reach**, **orbit**, compass, ambit]

9. Information Content: 3.1072
   Synset: [**year**, twelvemonth, **yr**]
   Synset: [**program**,**programme**]

As the result reveals, the topic's primary storyline is represented by the synset keywords: *space, land, earth, ground, discovery, state, nation, shuttle, center, station, plan, program, mission, concluding, final, last, day, international, land, astronaut, spaceman, orbit, reach, range, year, program.* This corresponds to the news topic's story, which reports about the space shuttle Discovery's final landing and retirement at the 9^{th} of March, 2011. Moreover, all news report that this mission is one of the last flights in the NASA space shuttle program.
All bold-faced synset-keywords do occur identically in the news text, while the plain-face words cannot be found. The first four merge steps (items #1 to #4) already reveal the most prominent keywords: *space, shuttle, discovery, earth, station* and *program.*
But, one can also see the already mentioned pitfalls resulting from wrong allocations from words to their senses. For example, merge step no. #5 assigns the word "*mission*" to the synset *[mission, missionary_post, missionary_station, foreign_mission]*, which deals with the religious aspect of evangelizing or converting one's religion. In addition to that one prominent fact is the filtering of proper nouns. It is inherited from the WordNet dependency, which incorporates only a very small amount of proper nouns. Consequently, proper nouns/phrases such as "John F. Kennedy Space Center" cannot be retrieved from WordNet and only "center" remains as a weak representative of this proper noun.

Now, that the main storyline key-synsets can be extracted, it is appropriate to uses them for retrieving the most storyline-compliant and substoryline-compliant articles. These articles can be found by means of the inverted index as noted above. In turn, these sub-clusters can be used as the data basis for an extractive summarization algorithm, such as LexRank or its topical-related summarization version (BoS LexRank).

4.4.4 News Recommendation

Figure 4.9 shows the main semantic view after its initialization.

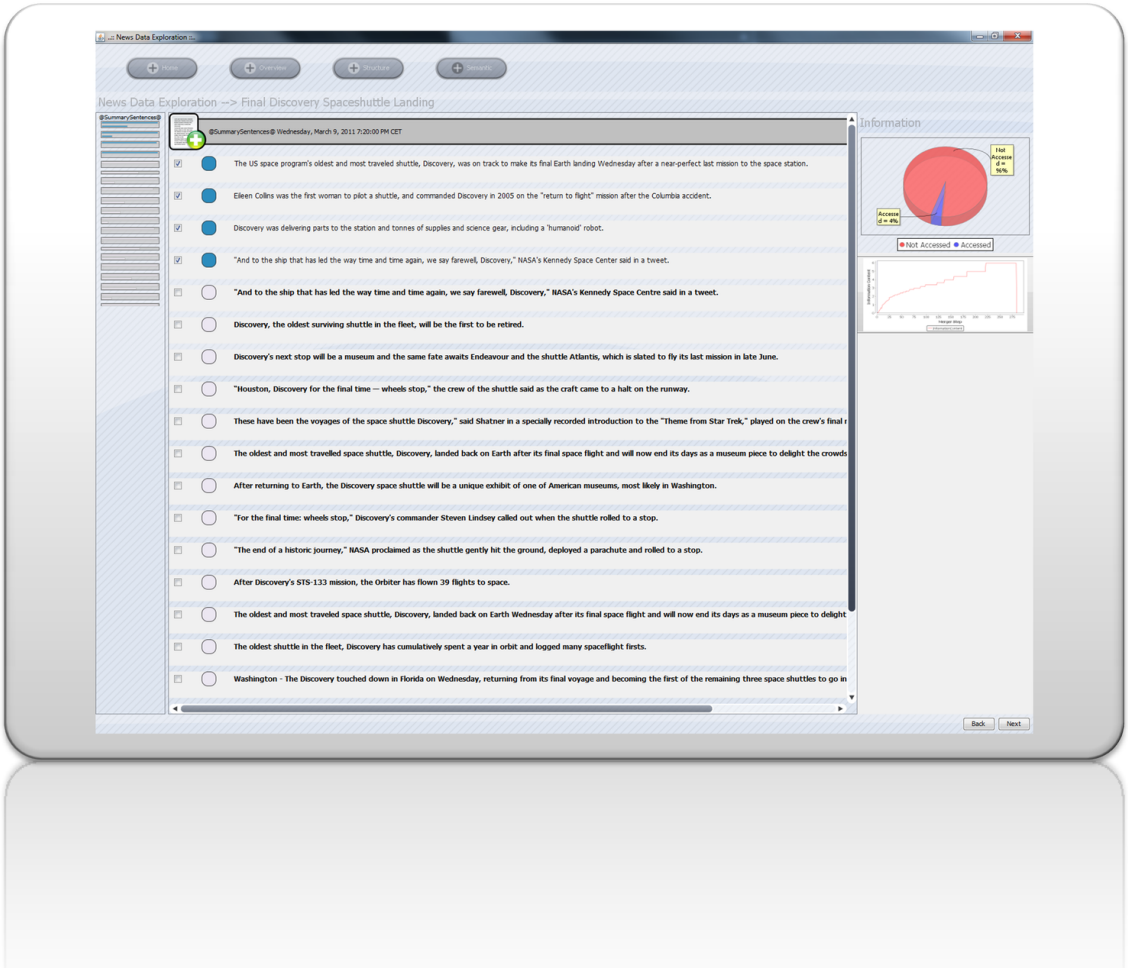


Figure 4.9: NewsGuide's Semantic View showing a Sentence-Based Topic Summary.

In comparison to the other views NewsGuide's semantic visualizations follows a different intuition than the previously mentioned views. Its key objective is to guide the users through the reading process and enhance their efficiency while reading. Thus, not the visualization stands in the primary focus, but rather the semantic-relatedness recommendation, which builds the foundation for the latter mentioned goal.
Figure 4.9 consists of three sub-components. First, the left-most history panel gives the user a visual representation of his reading process. Here, every document –represented by a document thumbnail– will be added if it was selected along the way. The thumbnail stays consistent with the already mentioned thumbnail implementation from Section 4.2.1, except of one aspect. The highlighting changes only between a dark and a light color, depicting read, respectively unread, items (sentences or paragraphs). Whenever the user selects one item as read, its coloring will change in the history accordingly.

The second component (middle panel) is the main textual screen. After the initialization it shows a sentence-based summary of the topic cluster. It was generated by taking only the main storyline-compliant documents into account and then ranking each article's sentences with a cosine-similarity based LexRank implementation. This has several advantages: While the first step allows to extract only articles, which are of interest for the primary storyline and thus abstracts away from the underlying textual data model, the second step retrieves only existing and well-formed salient sentences. Alternatively, one could also showcase the key-synset representation of the (sub-)storylines. Nevertheless, this would not be satisfying for the users, who expects to read full-text news articles. A further option could be to choose the most salient storyline-compliant document and present it to the user as a cluster representative. This –in turn– has the disadvantage that outstanding aspects, which are highlighted in several other documents could be omitted accidentally.
The text reader introduces also a relevance feedback mechanism into the process. Whenever the users read a paragraph they can mark it as read (i.e. the four top-most sentences in Figure 4.9). This leads to a series of actions after clicking the *Next* button at the bottom of the screen. All selected paragraphs –from the history and the currently read article– will embody the user's topic knowledge repository. In order to increase this repository effectively, the recommendation engine will retrieve articles, which were not read before and are semantically/topically different to the knowledge base. From the implementation point-of-view, NewsGuide maintains the knowledge repository in an artificial BoS vector. When a new recommendation is requested this vector will be compared to all unread articles. As a result a semantic similarity comparison is retrieved, which will be used in a graph-based importance ranking algorithm, such as LexRank (for the detailed functionality see Section 3.3.3). Taking the last five articles from the LexRank results allows to retrieve semantically unrelated articles. These articles can be seen as the ones with the highest information gain.
A further approach was also tested for the recommendation engine. Its idea was that an artificial document can be constructed from the read sentences. This document can be used as a reference point in the semantic comparison. For this purpose, the artificial knowledge repository document is compared to all other unread documents by means of the semantic DTW distance (see: Section 3.2.3 for a detailed description). Alike the BoS approach, the semantic relatedness scores are used in the LexRank implementation. While this approach can be seen as promising and leads to good results, it is immensely computationally intensive. Even in a ten document topic cluster its comparison step takes more than one hour to compute (with one read sentence). This computing time grows exponentially with the number of read sentences in the artificial document. Consequently, this approach had to be rejected not because of outstanding bad results, but its incomputability.
Figure 4.10 shows NewsGuide's approach for the visualization of the BoS recommendation results. It presents four most interesting articles in a document thumbnail

list.

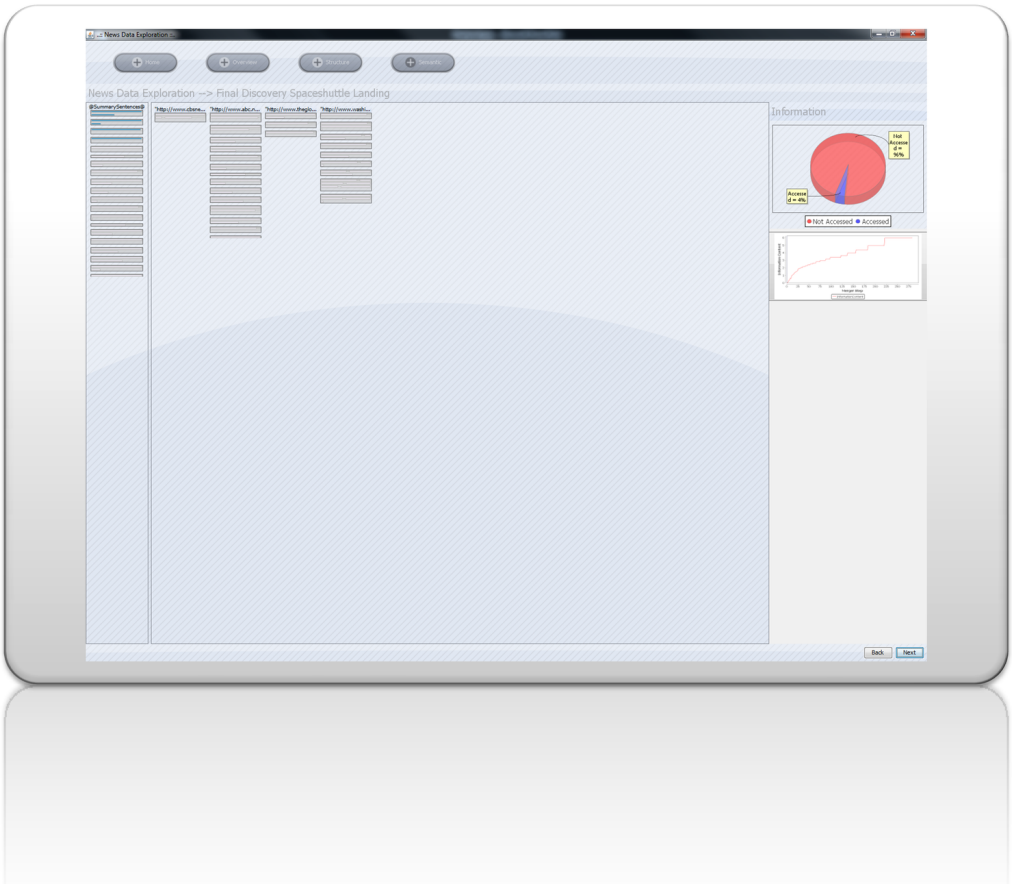


Figure 4.10: NewsGuide's Recommendation of Topically-Relevant News.

The recommendation technique enables the user to explore all mentioned (side-) aspects of the story in fast and efficient manner. In comparison to sophisticated relevance feedback mechanisms –such as the ones presented in Section 3.4– NewsGuide's relevance feedback mechanism is rather rudimentary. It forces the user mark every read paragraph. Yet still, its semantically inspired procedure is a new approach. It contrasts other text recommendation engines in such a way that it does not try to suggest articles, which match the user's intentional preferences, but recommends articles based on the already assessed knowledge base. If this knowledge repository could be gathered and harvested in the online news sphere it could lead to a revolutionary new reading experience.

The third component (right-most panel) is the information tracking screen. It gives the user an insight into his reading progress. It basically consists of two graph visualizations. The upper visualization is an intuitive pie chart. Its purpose is to reveal

the percentage of already accessed synset information.
The second graph is the information content graph, as presented in Section 4.4.3. Since NewsGuide is an interactive visual analytics tool, it tries to give the user as much freedom in choosing the variables as possible. Consequently, the component plots the information content graph in an interactive visualization. By clicking on the information content graph in the lower-right corner, a magnified and interactive version opens up. This dialog lets the user select an interval of merge steps and retrieves the list of articles that contain the selected synsets. The user can decide whether to investigate one of the retrieved articles more closely or select another interval range. Figure 4.11 shows the interactive graph visualization.

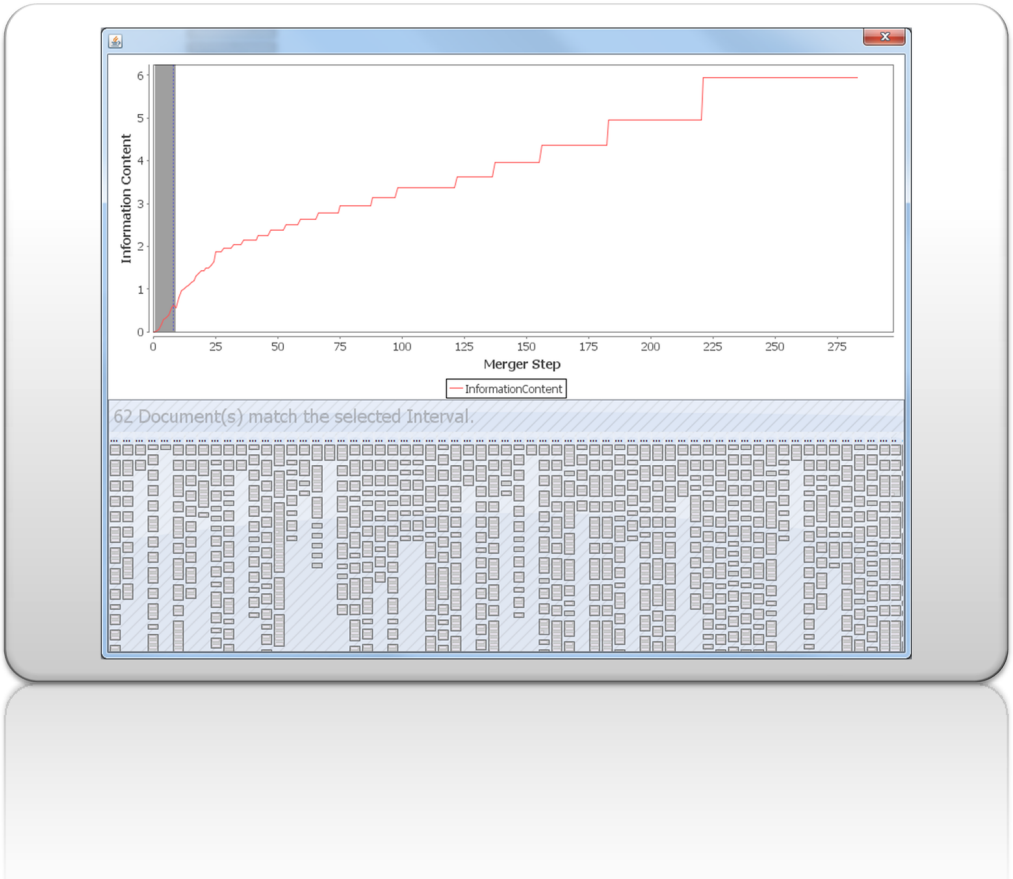


Figure 4.11: NewsGuide's Interactive Information Content Graph.

Summarizingly, NewsGuide introduces three new approaches towards an effective reading process. The first and most outstanding technique, deals with a topically-enhanced view onto the data. While the classical *Bag-of-Words* model only allows to compare documents on a strict word basis, its *Bag-of-Synset* enhancement compensates this drawback and allows to compare textual data on a semantic level. The BoS model

enables therefore to view the news cluster from a topic-related point-of-view. This technique lays the foundation for related more sophisticated analysis. One analysis step is the main storyline summarization, which extracts the topic's most storyline-compliant articles and extracts their primary plot. This overview is presented as an initial step in the reading process. It alleviates significantly the user's choice whether a topic is interesting or not.
Secondly, the main storyline extraction approach allows for a side-aspect oriented browsing. If the users want to see often or rarely occurring side-stories, they can explore the main storyline extraction plot for these articles and retrieve their attitudes and justifications.
Moreover, an information-gain effective news reading process stands in the focus of NewsGuide's recommendation engine. It allows the user to get a fast and accurate insight into the entire news cluster, while reducing the amount of redundantly accessed information.
All in all, NewsGuide is a multi-level news exploration system. It enables the user to see news article interrelations on the overview-level, text-correlations and patterns on the structural-level, and reveals non-obvious semantic viewpoints.

Chapter 5

Conclusion and Future Work

This chapter concludes the thesis with a summary of the contributions to the research in the field of news data analysis and proposes several topics that should be considered by future work.

5.1 Conclusion

This thesis had the overall goal to outline the research approaches in the field of news topic exploration. Therefore, an exemplary and abstract exploration pipeline was introduced, which contains all required steps for a profound news data analysis. The main approaches in each of the fields were described in-detail and put in context to the other techniques in the domain. In the project's initial research phase, the primary objectives were defined and recorded to function as an end-product requirement specification. The outstanding general target was the development of an interactive tool, which allow the users to investigative questions regarding overview, as well as in-depth analysis of a topic-coherent news cluster derived from multiple news sources. Based on this definition, we developed *NewsGuide*, a news exploration system for the mainstream news reader. It allows the user to develop an overview about a news topic, formulate hypothesis about non-obvious patterns, and gives means and techniques to prove or reject these hypothesis in a stream-lined visual interface. But, NewsGuide not only allows for a confirmatory analysis. It also enables users to do exploratory data analysis, which has to be seen as an undirected search for patterns in the data set. As one example, NewsGuide's structural view points the user to promising article comparisons. It uses standard visual data exploration techniques, such as heatmap color-coding of high text-/topic-interrelations to guide the reader to interesting facts.
NewsGuide not only uses and enhances existing data mining techniques, but it also develops new approaches, such as the knowledge repository recommendation engine or the main storyline extraction mechanism.

The main contributions and insights are summarized in the following paragraphs:

- NewsGuide enables the user to explore topic clusters on several abstraction levels. It allows the user to answer investigative questions on an overview-, structural- and semantic level. This multi-viewpoint approach is new for the news exploration field.
- NewsGuide lets the user investigated multiple mechanisms from measuring textual relatedness. Therefore, several textual similarity measures were implemented. The results reveal that even simple textual similarity measures perform well for answering inter-document relation patterns. One text-influence pattern is presented in Section 4.2.1.
- NewsGuide lets the user explore several semantic relatedness measures. Therefore, three different approaches in the field were implemented. This enables the user examine the semantic relatedness across documents. The results reveal that the semantic relatedness measures can be used effectively for a topical comparison across documents.
- It is noteworthy that nearly all semantic relatedness approaches are either extremely slow –i.e. the Semantic DTW measure described in Section 3.2.3– or reveal drastic implementation drawbacks, such as IP-related request limits in the case of the Google Distance Measure (described in Section 3.2.3) or the Pointwise Mutual Information - Information Retrieval approach of Turney (described in Section 3.2.3). In order to overcome these implementation and computation drawbacks, NewsGuide introduces the Bag-of-Synset model into the news exploration domain. This model proves to be an effective and efficient foundation on which sophisticated data analysis steps can be built.
- The Bag-of-Synset data representation builds on top of the classical Bag-of-Words model. It abstracts away from the words to a more generic and synonym-driven approach. Each word is represented by its synonym set, which incorporates all synonymous meanings of the word. Thus, by using a synonym-driven approach we can expect to find semantically related phrases, which could not have been found with the classical Bag-of-Words model.
- The Bag-of-Synset model can be retrieved in linear time, which makes it to an efficient semantically-enhanced data representation.
- NewsGuide introduces a new semantic relatedness measure on the Bag-of-Synset model. It is derived from correlation-coefficient techniques, which have proven to be successful in several application areas. It is one of the semantic similarity measures that the user can explore in the text-comparison view.

- The Bag-of-Synset similarity measure was additionally investigated as an enhancement to graph-based importance ranking algorithms, such as LexRank. It enhances the ranking of salient sentences with a semantically inspired point-of-view.

- NewsGuide develops a new approach towards a main storyline extraction. It is inspired by Resnik's information content approach (described in Section 3.2.3) and Huffman's greedy algorithm for extracting optimal prefix codes [Huf52].

5.2 Future Work

Along the development path several design decisions were made. These decisions are justified and put in comparison to the alternatives. Nevertheless, NewsGuide still has its limitations which can be –partly– tackled by further development- and research efforts. The following section will suggest possible extensions and improvements. Moreover, it will give hints for several lines of research arising from this work which could be pursued in future projects.

Word Sense Disambiguation
One of the primary improvements for the Bag-of-Synset (BoS) model will be the Word Sense Disambiguation, short WSD. This research field has several approaches for assigning the correct synset to a given word. Currently, only a very rudimentary WSD technique is implemented. Unsupervised or semi-supervised WSD techniques could be used to enhance the model. The WSD classification algorithm could incorporate the news cluster's available domain-knowledge to improve its precision rates.

Proper Noun Resolution
Another improvement for the NewsGuide system would be to incorporate named entity words and phrases. In the current implementation, which depends on WordNet for its BoS extraction, most named entity names are not available. Consequently, they had to be rejected as feature components in the vector. Yet still, all news articles are already leveraged with meta-data from the EMM pipeline (see: Section 2.1 for further information). Enhancing the BoS model with a meaningful, but not restrictive, named entity mechanism would preserve a great amount of semantics, which are now lost along the BoS extraction.

News Agency Inter-Relations
The NewsGuide structural view allows to explore textual inter-document relations. Hence, the next logical step would be to investigate patterns on the news agency level. Here, questions like "Which news agency is most prominent in a specific field?", or "Do

news agencies influence each other, and how?" could be answered. As a result, this field would have a much broader scope and would require a drastically larger data set than the one used in the current project. As a consequence, a myriad of computational problems would arise. Their problem description will be given in the next paragraph.

Incrementability and Scalability
Users are accustomed to a fluent news selection and reading process. However, this is not always possible if sophisticated data analysis steps are required to answer investigative questions. One possible solution is to pre-compute the solutions and present them to the user on-demand. Yet still, especially in interactive visualization this is often not possible. Moreover, the problem complexity grows steadily in the news domain, since –on the one hand– the amount of data increases constantly and –on the other hand– the number of possibilities continuously grows bigger. One solution for this would be to use or develop incremental algorithms for data analysis. One example is data stream clustering, which could be used for news exploration systems, such as the one developed in this project. Another possible solution would be to parallelize the computational steps and use the power of cloud-based computing.

Time-dependant Analysis
Time-dependency is an conspicuous factor particularly in the news domain. As one example, the classification of news always depends on a keyword-to-categories mapping. However, the amount of keywords increases over time and new important keywords arise with their corresponding news story, such as "smartphone" or "iPhone" (in the telecommunications segment). The question is how to deal with upcoming and unforeseen events programmically. This raises the question: "How can we teach algorithms to learn and also to forget?" A further example is news data clustering. Stories that report about the decisive goal in a soccer world-cup final have to be grouped into the world-cup topic of this year. Additionally, the world-cup topic has to be assigned as a subcluster into the soccer sector, which is once again a subcluster of sports. While this all seems obvious for a static reference point in time, it does ignore the fact that some topics arise (i.e. the next soccer world-cup) and others disappear from the news sphere (i.e. the third goal in the fourth soccer world-cup). Even more complex are cases in which news stories more and more merge over time into one story (i.e. a countries' peaceful protests change over to a civil war) or diverge (i.e. the tablet PC sector split off from the laptop sector).

News Cluster Evolution
A further future work field that is related to time-dependent analysis is news cluster evolution. Here, the question is if a news cluster's evolutionary development can be predicted. Trend analysis could help the user to see if a specific topic could become important over time. More specifically, one could investigated if news cluster develop-

ment stages can be related to the Gartner Hype Cycle (an instrument for measuring a technology's maturity; see [Gar11] for more information).

News Data Visualization
One question that relates to all data mining analysis steps is: "How can we convey the insights of our computations so that the user can easily understand them?" Visualizations prove to be very successful to guide the user in his understanding process. However, the text visualization domain is still far behind the numerical visualization domain. Particularly, the problem of scalable and incremental visualizations is not solved yet. Solutions in this field could be adapted directly for the news domain, in which a steadily flowing information stream awaits a meaningful visualization.

5.3 Summary

This master thesis had the goal to present all necessary steps for a sophisticated news data analysis. Therefore, it presented the main ideas and basic considerations in the introductory chapter 1. In order to understand the domain the entire value creation chain was described. Moreover, the introductory part presented the project's primary objectives and overall goals.
The Chapter 2 "Related Work" presented some of the most important, existing news data exploration systems. The EMM NewsExplorer, Lydia, NewsInsight and NewsBlaster were chosen to showcase the main approaches and techniques in this field.
The following Chapter 3 "News Data Exploration Techniques" described in its beginning an abstract news data exploration pipeline. Its main purpose is to emphasized, which pipeline steps can be improved and which already have sophisticated approaches and techniques. In the course of this first main chapter the techniques behind the news exploration pipeline are described. Therefore, text data clustering (see: Section 3.1), textual similarity measures (see: Section 3.2), text summarization techniques (see: Section 3.3), relevance feedback mechanisms (see: Section 3.4), information extraction methods (see: Section 3.5) and lastly text data visualization techniques (see: Section 3.6) were described functionally and with their relation to the news exploration domain.
The subsequent main Chapter 4 presented "NewsGuide", a news exploration system for the mainstream news reader. It tackles primary investigative questions regarding overview, and in-depth analysis of textual data. Therefore, it uses and enhances existing data mining techniques and develops new solutions for the problems, which emerged along the way. This chapter includes also a discussion of the design decisions. All advantages, disadvantages and alternatives are discussed, compared and evaluated.
Finally, the last Chapter 5.1 "*Conclusion and Future Work*" concluded the thesis with an enumeration of the main contributions and an outlook into the future of news exploration systems.

Bibliography

[ACD+98] Allan, J.; Carbonell, J.; Doddington, G.; Yamron, J.; Yang, Y.; Umass, J. A.; Cmu, B. A.; Cmu, D. B.; Cmu, A. B.; Cmu, R. B.; Dragon, I. C.; Darpa, G. D.; Cmu, A. H.; Cmu, J. L.; Umass, V. L.; Cmu, X. L.; Dragon, S. L.; Dragon, P. V. M.; Umass, R. P.; Cmu, T. P.; Umass, J. P.; Umass, M. S.: Topic Detection and Tracking Pilot Study Final Report. In *In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, S. 194–218, 1998.

[AGP+01] Agichtein, E.; Gravano, L.; Pavel, J.; Sokolova, V.; Voskoboynik, A.: Snowball: a prototype system for extracting relations from large text collections. In *SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data*. ACM Press, Juni 2001.

[Alt11] Altavista.com: Search engine - altavista.com. http://www.altavista.com/, 2011. [Online; accessed 25-July-2011].

[Ask11] Ask.com: Automated news aggregator - ask.com news. http://www.ask.com/news, 2011. [Online; accessed 25-July-2011].

[AWB03] Allan, J.; Wade, C.; Bolivar, A.: Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, S. 314–321. ACM, New York, NY, USA, 2003.

[BCS+07] Banko, M.; Cafarella, M. J.; Soderl, S.; Broadhead, M.; Etzioni, O.: Open information extraction from the web. In *In IJCAI*, S. 2670–2676, 2007.

[BE97] Barzilay, R.; Elhadad, M.: Using lexical chains for text summarization. In *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, S. 10–17, 1997.

[BGG+98] Bennett, C. H.; Gacs, P.; Gács, P.; Member, S.; Li, M.; Vitanyi, P. M. B.; Zurek, W. H.: Information distance. *IEEE Transactions on Information Theory*, Band 44, S. 1407–1423, 1998.

[Big11] BigNews.biz: Automated news aggregator - bignews.biz. http://bignews.biz/, 2011. [Online; accessed 25-July-2011].

[Ble11] Blei, D. M.: Introduction to probabilistic topic models. http://www.cs.princeton.edu/~blei/papers/Blei2011.pdf, 2011.

[BLP+04] Buntine, W.; Lofstrom, J.; Perkio, J.; Perttu, S.; Poroshin, V.; Silander, T.; Tirri, H.; Tuominen, A.; Tuulos, V.: A scalable topic-based open source search engine. In *Proc. IEEE/WIC/ACM Int. Conf. Web Intelligence WI 2004*, S. 228–234, 2004.

[BME99] Barzilay, R.; McKeown, K. R.; Elhadad, M.: Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, S. 550–557. Association for Computational Linguistics, Stroudsburg, PA, USA, 1999.

[BNJ03] Blei, D. M.; Ng, A. Y.; Jordan, M. I.: Latent dirichlet allocation. *J. Mach. Learn. Res.*, Band 3, S. 993–1022, March 2003.

[BO07] Bodenreider, O.; Olken, F.: Ontology summit 2007 communique. Online, April 2007.

[BP98] Brin, S.; Page, L.: The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, WWW7, S. 107–117. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1998.

[BP03] Banerjee, S.; Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness. In *Proc. of the 18th Int'l. Joint Conf. on Artificial Intelligence*, S. 805–810, 2003.

[Bre11] Brewer, C.: Heatmap coloring options - color brewer. http://colorbrewer2.org/, 2011. [Online; accessed 25-July-2011].

[Bri98] Brin, S.: Extracting patterns and relations from the world wide web. In *In WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, S. 172–183, 1998.

[BRS06] Best, C.; R., S.; S., H.: Web mining and intelligence - emm activity report 2005/2006. Online, 2006.

[CP04] Chklovski, T.; Pantel, P.: Verbocean: Mining the web for fine-grained semantic verb relations. In Lin, D.; Wu, D. (Hrsg.): *Proceedings of EMNLP 2004*, S. 33–40. Association for Computational Linguistics, Barcelona, Spain, July 2004.

[CS99] Collins, M.; Singer, Y.: Unsupervised models for named entity classification. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, S. 100–110, 1999.

[CV07] Cilibrasi, R. L.; Vitanyi, P. M. B.: The google similarity distance. *IEEE Trans. on Knowl. and Data Eng.*, Band 19, S. 370–383, March 2007.

[CVW09] Collins, C.; Viégas, F. B.; Wattenberg, M.: Parallel tag clouds to explore and analyze faceted text corpora. In *IEEE VAST*, S. 91–98. IEEE, 2009.

[Dic11] Dictionary, C. E.: Collins english dictionary - complete & unabridged 10th edition. http://dictionary.reference.com/browse/web2.0, Jul 2011.

[DUC11] DUC: Document understanding conferences - duc. http://www-nlpir.nist.gov/projects/duc/index.html, 2011. [Online; accessed 25-July-2011].

[ECD+04] Etzioni, O.; Cafarella, M.; Downey, D.; Kok, S.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D. S.; Yates, A.: Web-scale information extraction in knowitall, 2004.

[ECD+05] Etzioni, O.; Cafarella, M.; Downey, D.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D. S.; Yates, A.: Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, Band 165, S. 91–134, June 2005.

[Edm69] Edmundson, H. P.: New methods in automatic extracting. *Journal of the Association for Computing Machinery*, Band 16, S. 264–285, 1969.

[ER98] Elhadad, M.; Robin, J.: Surge: a comprehensive plug-in syntactic realization component for text generation. Technischer Bericht, Ben Gurion University of the Negev, 1998.

[ER04] Erkan, G.; Radev, D. R.: Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, Band 22, S. 457–479, December 2004.

[Eva03] Evans, R.: A framework for named entity recognition in the open domain. In *In Proceedings of the Recent Advances in Natural Language Processing (RANLP*, S. 137–144, 2003.

[Fac11] Facebook: Facebook social networking service. http://www.facebook.com/, 2011. [Online; Accessed 2011-03-28].

[Fil04] Filatova, E.: Event-based extractive summarization. In *In Proceedings of ACL Workshop on Summarization*, S. 104–111, 2004.

[Gar11] Gartner: Hype cycle technology maturity development - gartner. http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp, 2011. [Online; accessed 25-July-2011].

[GGRS96] Gilks, W.; Gilks, W.; Richardson, S.; Spiegelhalter, D.: *Markov chain Monte Carlo in practice*. Interdisciplinary statistics. Chapman & Hall, 1996.

[GHJV95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: *Design Patterns*. Addison Wesley, Reading, MA, 1995.

[GLYR07] Ghoniem, M.; Luo, D.; Yang, J.; Ribarsky, W.: Newslab: Exploratory broadcast news video analysis. In *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*, S. 123–130. IEEE Computer Society, Washington, DC, USA, 2007.

[Goo11] Google: Automated news aggregator - google news. http://news.google.com, 2011. [Online; accessed 25-July-2011].

[GS91] Garcia, M. R.; Stark, M. M.: *Eyes on the news*. St. Petersburg, Fla. : Poynter Institute for Media Studies, 1991.

[GS04] Griffiths, T. L.; Steyvers, M.: Finding scientific topics. *Proceedings of the National Academy of Science*, Band 101, S. 5228–5235, 2004.

[Has70] Hastings, W. K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, Band 57, Nr. 1, S. 97–109, April 1970.

[HBB10] Hoffman, M.; Blei, D.; Bach, F.: On-line learning for latent Dirichlet allocation. In *Neural Information Processing Systems*, 2010.

[HE02] Hamerly, G.; Elkan, C.: Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the eleventh international conference on Information and knowledge management*, CIKM '02, S. 600–607. ACM, New York, NY, USA, 2002.

[HHBL03] Holmqvist, K.; Holsanova, J.; Barthelson, M.; Lundqvist, D.: Reading or scanning? a study of newspaper and net paper reading. In *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, S. 657–670. Elsevier, 2003.

[HHN00] Havre, S.; Hetzler, B.; Nowell, L.: Themeriver: Visualizing theme changes over time. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS '00, S. 115–. IEEE Computer Society, Washington, DC, USA, 2000.

[HSO98] Hirst, G.; St-Onge, D.: Lexical chains as representation of context for the detection and correction malapropisms. citeseer.ist.psu.edu/hirst97lexical.html, 1998.

[Huf52] Huffman, D. A.: A method for the construction of minimum-redundancy codes. *Proceedings of the Institute of Radio Engineers*, Band 40, Nr. 9, S. 1098–1101, September 1952.

[HZZ03] Hoad, T. C.; Zobel, J.; Zobel, T. C. H. J.: Methods for identifying versioned and plagiarised documents. *Journal of the American Society for Information Science and Technology*, Band 54, S. 203–215, 2003.

[Ins85] Inselberg, A.: The plane with parallel coordinates. *The Visual Computer*, Band 1, Nr. 2, S. 69–91, August 1985.

[JC97] Jiang, J.; Conrath, D.: Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, S. 19–33, 1997.

[JG08] Ji, H.; Grishman, R.: Refining event extraction through cross-document inference. In *Proc*, 2008.

[JL09] Lin J.: Summarization. In M. "Ozsu T.; Liu L. (Hrsg.): *Encyclopedia of Database Systems*. Springer-Verlag, Heidelberg, Germany, 2009.

[JMF99] Jain, A. K.; Murty, M. N.; Flynn, P. J.: Data clustering: A review, 1999.

[JS03] Jarmasz, M.; Szpakowicz, S.: S.: Roget's thesaurus and semantic similarity. In *In: Proceedings of the RANLP-2003*, S. 212–219, 2003.

[Kha99] Kharitonov, M.: Cfuf: A fast interpreter for the functional unification formalism, 1999.

[KHDH02] Keim, D. A.; Hao, M. C.; Dayal, U.; Hsu, M.: Pixel bar charts: a visualization technique for very large multi-attribute data sets. *Information Visualization*, Band 1, S. 20–34, 2002.

[KKC94] Kwon, O.-W.; Kim, M.-C.; Choi, K.-S.: Query expansion using domain-adapted, weighted thesaurus in an extended boolean model. In *Proceedings of the third international conference on Information and knowledge management*, CIKM '94, S. 140–146. ACM, New York, NY, USA, 1994.

[KPC95] Kupiec, J.; Pedersen, J.; Chen, F.: A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '95, S. 68–73. ACM, New York, NY, USA, 1995.

[Kru09] Krums, J.: Hudson river crash. Twitter, Januar 2009.

[LC98] Leacock, C.; Chodorow, M.: Combining local context with WordNet similarity for word sense identification. In *WordNet: A Lexical Reference System and its Application*, 1998.

[Les86] Lesk, M.: 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*, SIGDOC '86, S. 24–26. ACM, New York, NY, USA, 1986.

[LH03] Lin, C.-Y.; Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, S. 71–78. Association for Computational Linguistics, Stroudsburg, PA, USA, 2003.

[LHZ+11] Lu, L.; Hou, L.; Zhang, X.; Wang, Z.; Li, J.: A case study of multi-view news exploration: Shanghai world expo. In *ACM WebSci'11*, S. 1–3, June 2011. WebSci Conference 2011.

[Lin98] Lin, D.: An information-theoretic definition of similarity. In *In Proceedings of the 15th International Conference on Machine Learning*, S. 296–304. Morgan Kaufmann, 1998.

[Lin11a] LingPipe: Natural language processing toolkit - lingpipe. http://alias-i.com/lingpipe/index.html, 2011. [Online; accessed 25-July-2011].

[Lin11b] LingPipe: Royalty free license - lingpipe. http://alias-i.com/lingpipe/licenses/lingpipe-license-1.txt, 2011. [Online; accessed 25-July-2011].

[LKS05] Lloyd, L.; Kechagias, D.; Skiena, S.: Lydia: A system for large-scale news analysis. In *Proceedings of String Processing and Information Retrieval (SPIRE)*, Nr. 3772 der Reihe Lecture Notes in Computer Science, S. 161–166, 2005.

[LLWL07] Liu, M.; Li, W.; Wu, M.; Lu, Q.: Extractive summarization based on event term clustering. In *Proceedings of the 45th Annual Meeting of the ACL on*

Interactive Poster and Demonstration Sessions, ACL '07, S. 185–188. Association for Computational Linguistics, Stroudsburg, PA, USA, 2007.

[LMB+06] Li, Y.; McLean, D.; Bandar, Z. A.; O'Shea, J. D.; Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. *IEEE Trans. on Knowl. and Data Eng.*, Band 18, S. 1138–1150, August 2006.

[Luh58] Luhn, H. P.: The automatic creation of literature abstracts. *IBM J. Res. Dev.*, Band 2, S. 159–165, April 1958.

[LV97] Li, M.; Vitanyi, P.: An introduction to kolmogorov complexity and its applications: Preface to the first edition, 1997.

[LYK+10] Luo, D.; Yang, J.; Krstajic, M.; Ribarsky, W.; Keim, D. A.: Eventriver: Visually exploring text collections with temporal references. *IEEE Transactions on Visualization and Computer Graphics*, Band 99, Nr. RapidPosts, S. 10–20, 2010.

[LZL06] Li, X.; Za
"ıane, O.; Li, Z.: *Advanced data mining and applications: second international conference, ADMA 2006, Xi'an, China, August 14-16, 2006 : proceedings*. Lecture Notes in Artificial Intelligence. Springer, 2006.

[Mar97] Marcu, D.: The rhetorical parsing, summarization, and generation of natural language texts. Technischer Bericht, University of Southern California, 1997.

[Mar11] Maramushi: News data visualization - newsmap project page. http://marumushi.com/projects/newsmap, 2011. [Online; accessed 25-July-2011].

[MB04] Moschitti, A.; Basili, R.: Complex linguistic features for text classification: A comprehensive study. *Advances in Information Retrieval*, Band 26, S. 181–196, 2004.

[MBC+05] Metzler, D.; Bernstein, Y.; Croft, W. B.; Moffat, A.; Zobel, J.: Similarity measures for tracking information flow. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, S. 517–524. ACM, New York, NY, USA, 2005.

[MBE+02] McKeown, K. R.; Barzilay, R.; Evans, D.; Hatzivassiloglou, V.; Klavans, J. L.; Nenkova, A.; Sable, C.; Schiffman, B.; Sigelman, S.; Summarization, M.: Tracking and summarizing news on a daily basis with columbia's newsblaster, 2002.

[MBF+90] Miller, G. A.; Beckwith, R.; Fellbaum, C.; Gross, D.; Miller, K. J.: Introduction to WordNet: An On-line Lexical Database*. *International Journal of Lexicography*, Band 3, Nr. 4, S. 235–244, Dezember 1990.

[Mel11] Melli, G.: Discriminative learning algorithm. http://www.gabormelli.com/RKB/Discriminative_Learning_Algorithm, 2011. [Online; accessed 25-July-2011].

[MG09] Muhr, M.; Granitzer, M.: Automatic cluster number selection using a split and merge k-means approach. In *DEXA Workshops*, S. 363–367, 2009.

[Min09] Minka, T.: Discriminative models, not discriminative training. Technischer Bericht, Technical Report MSR-TR-2005-144, Microsoft Research, October 2005. ftp://ftp. research. microsoft. com/pub/tr/TR-2005-144. pdf, 2009.

[MKH+99] McKeown, K. R.; Klavans, J. L.; Hatzivassiloglou, V.; Barzilay, R.; Eskin, E.: Towards multidocument summarization by reformulation: Progress and prospects. In *IN PROCEEDINGS OF AAAI-99*, S. 453–460, 1999.

[MMM06] Marneffe, M.; Maccartney, B.; Manning, C.: Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of LREC-06*, S. 449–454, 2006.

[Mon11a] Monitor, E. M.: Emm medisys - europe media monitor. http://medusa.jrc.it/medisys/homeedition/de/home.html, 2011. [Online; accessed 25-July-2011].

[Mon11b] Monitor, E. M.: Emm news brief - europe media monitor. http://emm.newsbrief.eu/, 2011. [Online; accessed 25-July-2011].

[Mon11c] Monitor, E. M.: Emm news explorer - europe media monitor. http://emm.newsexplorer.eu/, 2011. [Online; accessed 25-July-2011].

[MP98] Marsh, E.; Perzanowski, D.: Muc-7 evaluation of ie technology: Overview of results. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. http://www.itl.nist.gov/iaui/894.02/-related_projects/muc/index.html, 1998.

[Nag68] Nagy, G.: State of the art in pattern recognition. *IEEE J PROC*, Band 56, Nr. 5, S. 836–863, 1968.

[Nav09] Navigli, R.: Word sense disambiguation: a survey. *ACM COMPUTING SURVEYS*, Band 41, Nr. 2, S. 1–69, 2009.

[New11] NewsMap: News data visualization - newsmap. http://newsmap.jp/, 2011. [Online; accessed 25-July-2011].

[NSC08] Naughton, M.; Stokes, N.; Carthy, J.: Investigating statistical techniques for sentence-level event classification. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, S. 617–624. Association for Computational Linguistics, Stroudsburg, PA, USA, 2008.

[NTM06] Nadeau, D.; Turney, P. D.; Matwin, S.: Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *In 19th Canadian Conference on Artificial Intelligence*, 2006.

[OEC11] OECD: *News in the Internet Age: New Trends in News Publishing*. Organization for Economic Cooperation & Development, 2011.

[PC96] Park, Y. C.; Choi, K.-S.: Automatic thesaurus construction using bayesian networks. *Inf. Process. Manage.*, Band 32, S. 543–553, September 1996.

[Pew11a] Pew Research Center: How mobile devices are changing community information environments - pew research center. http://pewinternet.org/Reports/2011/Local-mobile-news.aspx, 2011. [Online; accessed 25-July-2011].

[Pew11b] Pew Research Center: The state of the news media 2011 - pew research center. http://stateofthemedia.org/2011/overview-2/, 2011. [Online; accessed 25-July-2011].

[Pew11c] Pew Research Center: Trends to watch news and information consumption - pew research center. http://www.slideshare.net/PewInternet/trends-to-watch-news-and-information-consumption, 2011. [Online; accessed 25-July-2011].

[Por97] Porter, M. F.: Porter stemmer. In Sparck Jones, K.; Willett, P. (Hrsg.): *Readings in information retrieval*, S. 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[Pri09] PricewaterhouseCoopers: Global entertainment and media outlook: 2009 - 2013. Online, 2009.

[PS07] Ponzetto, S. P.; Strube, M.: Knowledge derived from wikipedia for computing semantic relatedness. *J. Artif. Int. Res.*, Band 30, S. 181–212, October 2007.

[Rad01] Radev, D. R.: Experiments in single and multidocument summarization using mead. In *In First Document Understanding Conference*, 2001.

[Res93] Resnik, P. S.: Selection and information: A class-based approach to lexical relationships. Technischer Bericht, University of Pennsylvania, 1993.

[Res95] Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, S. 448–453, 1995.

[RHM99] Radev, D. R.; Hatzivassiloglou, V.; Mckeown, K. R.: A description of the cidr system as used for tdt-2. In *In DARPA Broadcast News Workshop*, 1999.

[RJST04] Radev, D. R.; Jing, H.; Sty, M.; Tam, D.: Centroid-based summarization of multiple documents. *Inf. Process. Manage.*, Band 40, S. 919–938, November 2004.

[RLC+06] Rothwell, R.; Lehane, B.; Chan, C. H.; Smeaton, A. F.; E, N.; Jones, G. J. F.; Diamond, D.: The cdvplex biometric cinema: Sensing physiological responses to emotional stimuli in film. Advances in Pervasive Computing 2006, May 2006.

[RMBB89] Rada, R.; Mili, H.; Bicknell, E.; Blettner, M.: Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, Band 19, Nr. 1, S. 17–30, Januar 1989.

[Ros10] Ross, S.: *A first course in probability.* Pearson Prentice Hall, 2010.

[SB88] Salton, G.; Buckley, C.: Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, Band 24, Nr. 5, S. 513–523, 1988.

[Sch98] Schütze, H.: Automatic word sense discrimination. *Comput. Linguist.*, Band 24, S. 97–123, March 1998.

[SG06] Steyvers, M.; Griffiths, T.: Probabilistic topic models. In Landauer, T.; Mcnamara, D.; Dennis, S.; Kintsch, W. (Hrsg.): *Latent Semantic Analysis: A Road to Meaning.* Laurence Erlbaum, 2006.

[SH06] Sahami, M.; Heilman, T. D.: A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, S. 377–386. ACM, New York, NY, USA, 2006.

[Sin91] Sinclair, J.: *Corpus, concordance, collocation.* Describing English language. Oxford University Press, 1991.

[SNMS02] Schiffman, B.; Nenkova, A.; McKeown, K.; Summarization, M.: Experiments in multidocument summarization, 2002.

[Sus97] Sussna, M. J.: *Text retrieval using inference in semantic metanetworks*. Dissertation, University of California at San Diego, La Jolla, CA, USA, 1997. UMI Order No. GAX97-26031.

[Tex11a] Textfixer: English stop word list - textfixer. http://www.textfixer.com/resources/common-english-words.txt, 2011.

[Tex11b] TextMap: Textmap - lydia project's web frontend. http://www.textmap.com/, 2011. [Online; accessed 25-July-2011].

[Tim11] Times, N. Y.: Online news portal - new york times. http://global.nytimes.com//, 2011. [Online; accessed 25-July-2011].

[Tur01] Turney, P. D.: Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, S. 491–502. Springer-Verlag, London, UK, 2001.

[Twi11] Twitter: http://twitter.com/ - social networking service, 2011. [Online; Accessed 2011-03-28].

[VHSZ09] V., C.; Hasan; Salem; Zaki: SPARCL: An effective and efficient algorithm for mining arbitrary shape-based clusters. *Knowledge and Information Systems*, Band 21, Nr. 2, S. 201–229, Nov 2009. Invited: best papers of ICDM'08.

[Wan07] Wang, Y.: Distributed Gibbs Sampling of Latent Dirichlet Allocation : The Gritty Details. Technischer Bericht, Tsinghua University, 2007.

[Wik11] Wikipedia: Rss - wikipedia, the free encyclopedia. //en.wikipedia.org/w/index.php?title=RSS&oldid=448485621, 2011. [Online; accessed 28-September-2011].

[Wil08a] Wilkens, M.: Evaluating pos taggers: Precision. http://mattwilkens.com/2009/01/26, January 2008. [Online; accessed 25-July-2011].

[Wil08b] Wilkens, M.: Evaluating pos taggers: Speed. http://mattwilkens.com/2008/11/08/, 2008. [Online; accessed 25-July-2011].

[Wor11] WordNet: Wordnet taxonomy. http://www.srcf.ucam.org/~rb432/bergmair.cjb.net/htdocs/pub/towlingsteg-rep-inoff-web.www/node8.html, 2011. [Online; accessed 25-July-2011].

[WP94] Wu, Z.; Palmer, M.: Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, S. 133 –138. New Mexico State University, Las Cruces, New Mexico, 1994.

[Yah11] Yahoo: Automated news aggregator - yahoo news. http://news.yahoo.com/, 2011. [Online; accessed 25-July-2011].

[YHTT09] Yaman, S.; Hakkani-Tür, D.; Tür, G.: Combining semantic and syntactic information sources for 5-w question answering. In *INTERSPEECH*, S. 2707–2710, 2009.

[YMM09] Yao, L.; Mimno, D.; McCallum, A.: Efficient methods for topic model inference on streaming document collections. In *KDD*, 2009.

[ZH05] Zhou, X.; Han, H.: Survey of word sense disambiguation approaches. In *Proceedings of the 18th International Florida AI Research Society Conference*, 2005.

[Zho05] Zhong, S.: Efficient online spherical k-Means clustering. In *Proc. 2005 IEEE International Joint Conference on Neural Networks*, S. 3180–3185, 2005.

[ZKL08] Zhang, H.; Koskela, M.; Laaksonen, J.: Report on forms of enriched relevance feedback. deliverable d1.1 of fp7 project nº 216529 pinview. Technischer Bericht Nr. 10; TKK-ICS-R10, Teknillinen korkeakoulu, Espoo, 2008.

[ZNL+09] Zhu, J.; Nie, Z.; Liu, X.; Zhang, B.; Wen, J. R.: StatSnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th international conference on World wide web*, WWW '09, S. 101–110. ACM, New York, NY, USA, 2009.

[ZSWH11] Zhang, J.; Sun, Y.; Wang, H.; He, Y.: Calculating statistical similarity between sentences. In *JCIT: Journal of Convergence Information Technology*, S. 22 - 34, 2011.

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und nur mit erlaubten Hilfsmitteln angefertigt habe.

Konstanz, den September 28, 2011

Michael Behrisch