Automatically Identifying Lexical Chains by Means of Statistical Methods—A Knowledge-Free Approach

Automatisches identifizieren lexikalischer Ketten mittels statistischer Methoden—Ein vorwissensfreier Ansatz

Master-Thesis von Steffen Remus
Oktober 2012

TECHNISCHE
UNIVERSITÄT
DARMSTADT

UBIQUITOUS
KNOWLEDGE
PROCESSING

Automatically Identifying Lexical Chains by Means of Statistical Methods—
A Knowledge-Free Approach

Automatisches identifizieren lexikalischer Ketten mittels statistischer Methoden—
Ein vorwissensfreier Ansatz

Master-Thesis
Eingereicht von Steffen Remus

1. Gutachten: Prof. Dr. Chris Biemann
2. Gutachten: Dr. Sabine Bartsch

Tag der Einreichung:

Technische Universität Darmstadt
Department of Computer Science

Ubiquitous Knowledge Processing (UKP) Lab

Erklärung zur Master-Thesis

Hiermit versichere ich die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 24. Oktober 2012

(S. Remus)

Abstract

The identification of lexical chains is an important building block in modern natural language processing applications such as summarization or text-segmentation. In order to extract lexical chains it is is necessary to identify the important lexico-semantic relations in a text.

Traditional approaches mostly rely on the use of knowledge resources such as thesauri or lexical databases like WordNet. Hence, the quality of the extracted lexical chains highly depends on the quality and quantity of the entries in the particular knowledge resource. Statistical methods on the other hand have proven to deliver good results in many natural language processing applications where the only prerequisite is a sufficiently large and qualitatively good data collection.

This thesis examines the suitability of statistical methods for the task of identifying lexico-semantic relations in order to build proper lexical chains. Four algorithms are developed that utilize either *latent Dirichlet allocation* (LDA) as a probabilistic topic modeling framework or the *log-likelihood ratio* (LLR) as an indicator for statistically significant co-occurring terms.

An intrinsic evaluation of these algorithms against some trivial and established baseline algorithms is performed which confirms the hypothesis. For this purpose a secondary issue of this thesis is the manual annotation of lexical chains and the development of a suitable lexical chain comparison measure. Supporting annotation guidelines are developed and an annotation toolkit is adapted in order to carry out an annotation project where two annotators annotated around 100 documents from the Salsa 2.0 corpus.

Further an extensive survey in the domain of clustering measures is performed in order to find a proper measure for the evaluation of the algorithms as well as for the quantification of the inter-annotator agreement. This measure is finally built of a combination of the *adjusted Rand index* (ARI) and the *normalized basic merge distance* (NBMD).

Supplemental material can be found at
http://www.ukp.tu-darmstadt.de/data/lexical-chains-for-german

Zusammenfassung

Das Extrahieren von lexikalischen Ketten ist heutzutage ein wichtiger Bestandteil moderner sprachverarbeitender Applikationen wie z.B. automatische Textzusammenfassung oder Textsegmentierung. Das erfolgreiche Extrahieren lexikalischer Ketten setzt die Identifikation von aussagekräftigen lexikalisch-semantischen Relationen in einem Text voraus.

Traditionelle Ansätze greifen hierzu auf Wissensressourcen wie Thesauri oder lexikalische Datenbanken zurück. Folglich ist die Qualität der resultierenden lexikalischen Ketten im hohen Maße abhängig von der Qualität und Quantität der Einträge in der Wissensressource. Statistische Methoden hingegen liefern erwiesenermaßen gute Ergebnisse in vielen sprachverarbeitenden Anwendungsgebieten mit einer hinreichend großen und qualitativ guten Textsammlung als einziger Voraussetzung.

Diese Arbeit untersucht die Tauglichkeit von statistischen Methoden für die Identifikation lexikalisch-semantischer Relationen und somit lexikalischer Ketten. Vier Algorithmen wurden entwickelt welche die Hypothese sinnvoll unterstützen sollen. Diese Algorithmen basieren entweder auf *latent Dirichlet allocation* (LDA), einem probabilistischem Topic Modeling Framework, oder auf dem *log-likelihood ratio* (LLR) welches ein Indikator für statistisch signifikante gleichzeitig auftretende Wörter ist.

Die Hypothese wird anhand einer intrinsisch durchgeführten Evaluation bestätigt. Deshalb beschäftigt sich diese Arbeit weiterhin mit der manuellen Annotation von lexikalischen Ketten sowie der Entwicklung eines geeigneten Vergleichsmaßes. Annotationsrichtlininen wurden entwickelt und ein Annotationsprogramm wurde adaptiert um ein Annotationsprojekt zu unterstützen in dem zwei Annotatoren c.a. 100 Dokumente des Salsa 2.0 Korpus annotierten.

Weiterhin wurden diverse Evaluationsmaße der Clustering Domäne ausführlich untersucht um ein angemessenes Evaluations- sowie Inter-Annotator Übereinstimmungsmaß auszuwählen. Dieses setzt sich letztendlich aus einer Kombination des *adjusted Rand index* (ARI) und der *normalized basic merge distance* (NBMD) zusammen.

Zusätzliches Material ist verfügbar unter
http://www.ukp.tu-darmstadt.de/data/lexical-chains-for-german

Acknowledgments

I want to sincerely thank Chris for giving me the opportunity to write this thesis and for being a great knowledge resource, supervisor, annotator and motivator in doubtful times. I really appreciate all his efforts for guiding me trough this work.

I also want to thank Richard and Thomas for their support and belief in my work.

Next I want to thank all my friends that kept patient with me while I was struggling so hard.

Meinen Eltern Petra und Peter möchte ich zutiefst dafür danke das sie mich in allen Lebenslagen unterstützten und immer ein offenes Ohr für mich hatten.

Last but not least I want to express my deepest love and gratitude to Katrin who was always there when she was needed most. She remembered me to live my life apart from desk to clear my thoughts in times they got stuck.

Contents

1 Introduction

The hunger of humanity for information is endless. If it is for work, school, or private interest, we seek for information nearly every minute. But the human processing of the enormous mass of information which is entailed in large collections of unstructured text like the World Wide Web is simply impossible. Even the amount of new or changing information that comes in every day or even every hour is tremendously large. The reliable support by computer systems is indispensable nowadays.

While some information is processable well in an automatic way due to its structuredness like cartographic data or the like, other is still embedded in written or spoken text. It is a major part of modern natural language processing research to computationally understand a text and extract structured from unstructured data.

A text that is understandable by its nature exhibits an underlying structure which makes the text coherent; that is, the structure is responsible for making the text "hang" together (Halliday and Hasan 1976). The theoretic foundation of this structure is defined as *coherence* and *cohesion*. While the former is concerned with the meaning of a text, the latter can be seen as a collection of devices for creating it. Cohesion and coherence build the basis for most of the current natural language processing problems that deal with text understanding simply because a natural precondition is a coherent and cohesive text.

Currently only cohesion qualifies for the automatic analysis due to the processing of text without the need of considering the situational environment. Different tasks in natural language processing require either explicit or implicit analysis of different cohesive features. For example co-reference resolution is typically addressed by the analysis of the cohesive devices of *reference* and *substitution* while the problem of e.g. word sense disambiguation is typically addressed by the cohesive devices of *lexical cohesion*.

Lexical cohesion ties together words or phrases that are semantically related. For example the words "book" and "novel" are related because "book" is a hypernym of "novel". Though, the automatic analysis is a tough problem that involves not only the identification of a semantic relation but also the disambiguation of multiple possible relations and thus the weighting of some kind of significance for the current text. For example, is the word "novel" related to "thesis" because they are both kinds of writing?—the answer depends on the overall purpose of the text.

Once all the cohesive ties are identified the involved items can be grouped together to form so-called *lexical chains*. For example when having a cohesive tie involving the words "car" and "tire" and another tie involving the words “car” and “brake” we can build a lexical chain containing the words “car”, “tire”, and “brake” because they form a cohesively connected unit. By doing so, lexical chains contain only expressions that refer to the same concept or topic.

Lexical chains thus form a theoretically well founded building block in various natural language processing applications. They are used for example in word sense disambiguation (Okumura and Honda 1994), summarization (Barzilay and Elhadad 1997), malapropism detection & correction (Hirst and St-Onge 1998), document hyperlinking (Green 1996), text segmentation (Stokes et al. 2004), topic tracking (Carthy 2004), and many more. Here, the performance of the individual task mainly depends on the quality of the identified lexical chains.

1.1 Motivation

Current approaches mainly focus on the use of knowledge resources like lexical semantic databases (Hirst and St-Onge 1998) or thesauri (Morris and Hirst 1991) as background information in order to resolve possible semantic relations for the use of lexical chaining. This strategy has some serious drawbacks:

1. The quality of lexical chains highly depends on the quality of the resource. This includes the number of terms or phrases that are included in the database as well as the number and the kind of relations between those entries.
2. Resources may be of low quality for rare registers or special domains like the biomedical domain or even for resource-scarce languages like Swahili.
3. The use of certain resources may be limited due to policy restrictions or the resource may be only available online which reduces the computational efficiency.
4. The structure of specific resources may limit the connectivity between certain kinds of items. E.g. lexical databases relate terms based on grammatical features like synonymy and group them by their part-of-speech. Grammatical relations across different parts-of-speech occur much less frequently than relations within the same part-of-speech and also the number of relations for nouns is much higher than for any other part-of-speech. Hence many algorithms that use lexical databases limit their search scope to nouns only.
5. The costs-benefit ratio for manually compiling such a resource is unbalanced; the costs are simply too expensive.

Hence the choice of the resource has a direct impact on the resulting lexical chains and thus on the performance of the certain task.

Statistical methods on the other hand have proven to deliver good results in many natural language processing applications. *Topic models* for example define a probabilistic framework for dimensionality reduction. The intuitive interpretation is that a document is a composition of topics and a topic is a composition of terms. A topic thus captures the semantics of a concept and as a result terms can be mapped to meanings. Topic models are already used for tasks such as summarization (Gong and Liu 2001; Hennig 2009), text segmentation (Misra et al. 2009; Riedl and Biemann 2012), lexical substitution (Dinu and Lapata 2010), word sense disambiguation (Cai et al. 2007; Boyd-Graber et al. 2007), etc.

Topic models typically operate on a *term-document* matrix and thus on the bag-of-words representation of documents ignoring the order of terms. Other statistical methods for natural language processing operate on different input data, e.g. the log-likelihood ratio is a significance measure that is typically used with word co-occurrences, hence a term-term matrix. Natural language processing tasks where word co-occurrences are successfully used are for example summarization (Lin and Hovy 2000), keyphrase detection (Tomokiyo and Hurst 2003), collocation analysis (Manning and Schütze 1999) and a variety of unsupervised structure discovery techniques as described in Biemann (2012).

The only prerequisite of statistical methods is a sufficiently large data collection containing texts of a good and uniform quality.

1.2 Hypothesis

Lexical chains are a convenient intermediate representation of lexical cohesion and cohesion is an indicator for the strength of unity in text. Techniques that constantly identify reliable lexical chains allow the development of higher level algorithms dealing with natural language processing tasks using a theoretically well founded infrastructure.

The fact that statistical methods are already used for the completion of natural language processing tasks which implicitly assume lexical cohesion gives rise to the following question which is the central theme in this thesis:

> Can we use statistical methods for the extraction of lexical chains that are qualitatively as good as or even better than lexical chains extracted with the help of knowledge resources?

1.3 Outlook

The thesis is structured as follows: In Chapter 2 the concept of lexical chains, as well as a selection of established and current state-of-the-art algorithms for lexical chain extraction are described, some of which will be used in the evaluation as baseline algorithms.

Most of the current lexical chain extraction techniques are evaluated extrinsically, which means that a certain natural language processing problem is addressed by utilizing lexical chains and the quality of the lexical chaining algorithm is measured by evaluating the specific task. While this strategy is eligible a secondary issue of this thesis is the development of a methodology for the intrinsic evaluation of lexical chains. Hence, the hypothesis will be validated by directly comparing automatically extracted lexical chains with manually annotated lexical chains which can be seen as the gold standard.

Chapter 3 addresses the problem of finding a proper lexical chain comparison measure suited for inter-annotator agreement as well as for the final evaluation. Since the structure of a set of lexical chains from a certain document is somewhat similar to the structure of a mathematical clustering, but unfortunately the perfect clustering measure does not exist, an extensive survey will be performed that tests a number of measures from the clustering domain for their suitability of comparing sets of lexical chains. A combination of some of the tested measures is then chosen which best fulfills a number of defined properties.

Chapter 4 describes the development of human annotated lexical chains. Therefore, the general problem of subjectivity in the interpretation of a text while annotating lexical chains is discussed and the development of supporting annotation guidelines as well as the adaption of an annotation toolkit is presented here. Also, the agreement of the human annotated lexical chains will be analyzed using the measure from Chapter 3 and it will be shown that subjectivity in interpretation is still a major problem one has to be aware of.

In order to support the central hypothesis of this work, chapter 5 develops four statistical approaches for lexical chaining by considering two branches of statistical natural language processing. Three methodologies will utilize *latent Dirichlet allocation* (LDA) as a probabilistic topic modeling framework each using the information provided by LDA differently, and one methodology will make use of the *log-likelihood ratio* (LLR) in order to identify statistical significant word co-occurrences.

The validation of the hypothesis will be described in chapter 6. Here, the statistical methods will be intrinsically evaluated against various baseline algorithms using the manually annotated lexical chains from Chapter 4 and the lexical chain comparison measure from Chapter 3. It will be shown that in our setting lexical chaining algorithms based on statistical methods outperform knowledge resource based algorithms.

Chapter 7 finally summarizes and concludes this thesis, discusses the developed lexical chain extraction algorithms and provides some future directions.

2 Lexical Chains

Lexical chaining is a means for making the instantiation of *lexical cohesion* in a *text* explicit. That being said, the terms *lexical cohesion*, *cohesion* itself, and *text* must be further defined in order to fully describe the tool of *lexical chains* as a whole. This chapter clarifies the questions what is *cohesion*, what do *lexical chains* have to do with it, and what gain do we get when we extract them?

2.1 The Conception of Cohesion

Most of the upcoming definitions are taken from the work *Cohesion in English* by Halliday and Hasan (1976), who mainly formed the conception of cohesion as it is today.

Cohesion is the abstract force that makes a *text* a text. It is implicitly embodied in every text we read or write, as long as we can say the text is not simply nonsense, i.e. it is not just a collection of some random sentences. This is best described with an example, consider therefore the sentences below:

(2.1) It was all very well to say 'Drink me,' but the wise little Alice was not going to do THAT in a hurry.

(2.2) However, this bottle was NOT marked 'poison,' so Alice ventured to taste it.

(2.3) In Barcelona it is raining today.

One can easily determine that the example sentences 2.1 and 2.2 are somehow related and belong to the same text whereas sentence 2.3 clearly falls out of scope and does not fit into the current environment. Although the structure of 2.3 is correct and it definitely bears some meaning, it is not about the same thing as 2.1 and 2.2.

Halliday and Hasan (1976) defined sentences forming a unified whole, i.e. sentences being about the same thing, interpretable to deliver a message, to have *texture*, and thereby to be *text*. Texture is thus the property that every text has per definition; it can be seen as the organization of text.

Structure, on the other hand, is a natural requirement for achieving texture. Sentences are the biggest grammatical unit and it is assumed that they express some meaning which implies, that sentences are the smallest unit of meaning and thus a text is no more a unit of grammar it is a unit of meaning.

Cohesion and *coherence* in turn are responsible for creating texture. While the latter will not be discussed here, we will focus on the former. *Cohesion* sticks or ties together one *lexical item* — which may be a word, an expression, or a phrase — with another presupposed lexical item that has gone before. This link is then called a *cohesive tie* and it is not bound to a certain sentence. More it acts across (multiple) sentences, which has then the effect of making these sentences cohere with each other.

As an illustration consider the expression “Drink me” from the example sentence 2.1 and the word “THAT” from the same sentence. The reader simply knows these two are related; without the former, the latter can not be resolved. This example shows that the term *tie* refers to a pair of items that are connected by some *semantic relation* defined by cohesion where one item always provides the source of interpretation for the other item.

The semantic relation also called *cohesive relation* can take one of many forms. Halliday and Hasan described a cohesive relation to be either *grammatical* or *lexical*; i.e. resolvable either by form or by lexis. Types of grammatical cohesion are: i.) Reference, ii.) Substitution, iii.) Ellipsis, and iv.) Conjunction (which is partially also lexical but will not be covered by this thesis). However, this thesis addresses only lexical cohesion as the dominant number of cohesive ties is instantiated by lexical cohesion (Hasan 1984; Hoey 1991).

Lexical cohesion creates cohesion only by the choice of the vocabulary. The main types of *lexical cohesion* as described by Halliday and Hasan are: i.) *general noun*, ii.) *repetition*, iii.) *synonymy* and *near-synonymy*, iv.) *superordinate term*, and v.) *collocational*, where the name *collocational* is a rather bad choice (Hoey 1991), since the term *collocation* is mainly used to refer to statistically significant term co-occurrence. Morris and Hirst (1991) called this type of semantic relation a *general association of ideas* which sounds broader but better reflects the situation.

Halliday and Hasan noted that lexical items which are tied with other lexical items which are again tied with again other lexical items and so on form a so-called *chain*. Yet, the term *lexical chain* was first mentioned by Hasan where she described it to come “. . . closest to the realization of some part of a semantic field” (Hasan 1984, p.187). Because of difficulties in the analysis of lexical cohesion in her work — note that the term *collocational* as used by Halliday and Hasan (1976) is not well defined; the boundaries between what is a collocational tie and what is not, are not clearly set — Hasan divided lexical chains into two categories: *identity chains*, which are instantiated through co-referentiality and thus are only interpretable in the context of the text, and *similarity chains*, which are instantiated through co-classification and co-extension and thus having a “language-wide validity” (Hasan 1984, p.201). Thereby she redefined the categories of *lexical cohesion* and eliminated the collocational category. She also noted that both types of chains are necessary in a normal non-minimal text. Furthermore Hasan introduced the notion of *chain interaction* which makes the individual chains linkable to other chains via their items. Putting this all together finally results in the concept of *cohesive harmony*.

The exact definition of *lexical chains* as it is generally used today was introduced by Morris and Hirst (1991) in the context of a computational approach for extracting lexical chains. In their work they defined a *lexical chain* to be “. . . a succession of a number of nearby related words spanning a topical unit of the text.” (p.22), which is nearly equivalent to the notion of similarity chains by Hasan (1984).

As an example consider the text in 2.4 which shows an extract from “Alice in Wonderland” by Lewis Carroll and some identified lexical chains. Although it is just one sentence — remember that searching for lexical chains within a single sentence is rather pointless in practice since a sentence itself is already a cohesive unit — the text comprises six lexical chains. The superscript numbers refer to the position of the respective word in the text.

(2.4) Alice^{1} was beginning to get very tired^{7} of sitting^{9} by her sister^{12} on the bank^{15}, and of having^{18} nothing^{19} to^{20} do^{21}: once^{22} or twice^{24} she had peeped^{27} into the book^{30} her sister^{32} was reading^{34}, but it had no pictures^{39} or conversations^{41} in it, ’and what is the use of a book^{51},’ thought Alice^{53} ’without pictures^{55} or conversation^{57}?’

1: { Alice^{1}, Alice^{53} }

2: { tired^{7}, having^{18} nothing^{19} to^{20} do^{21} }

3: { sister^{12}, sister^{32} }

4: { sitting^{9}, bank^{15} }

5: { once^{22}, twice^{24} }

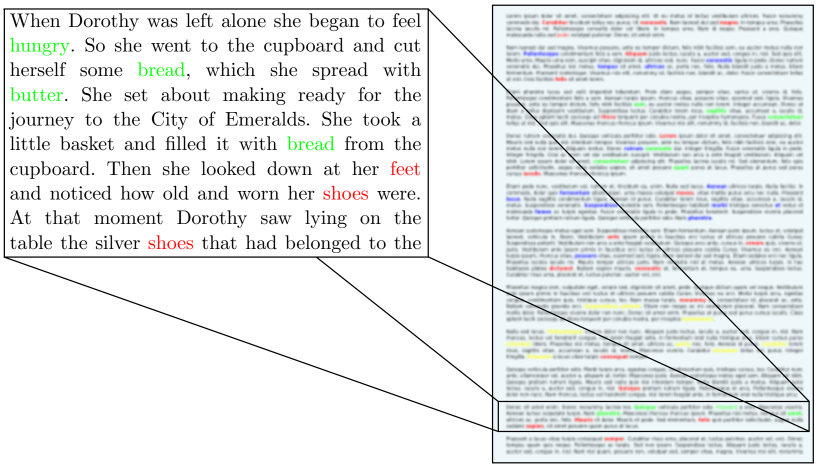
6: { peeped^{27}, book^{30}, reading^{34}, pictures^{39}, conversations^{41}, book^{51}, pictures^{55}, conversation^{57} }

From the six identified lexical chains, five consist only of a single tie. Note that in chain 2 the word “tired” forms a cohesive tie with the whole expression “having nothing to do”. Chain 6 is made up of various single ties:

1. peeped^{27} $\xleftrightarrow{\textit{co-hyponymy}}$ reading^{34}
2. book^{30} $\xleftrightarrow{\textit{repetition}}$ book^{51}
3. pictures^{39} $\xleftrightarrow{\textit{repetition}}$ pictures^{55}
4. conversations^{41} $\xleftrightarrow{\textit{repetition}}$ conversation^{57}
5. book^{30} $\xleftrightarrow{\textit{collocational}}$ reading^{34}
6. book^{30} $\xleftrightarrow{\textit{collocational}}$ pictures^{39}
7. book^{30} $\xleftrightarrow{\textit{collocational}}$ conversations^{41}

Note that the ties 5, 6 and 7 represent only three of ten ties regarding the repetitions of “book”, “picture” and “conversation”, the others are omitted for brevity.

Lexical chains are an important tool for *computational text understanding systems* because they are computable without requiring a deep understanding of the text (Barzilay 1997). Further they provide a context for the resolution of ambiguity of word instances to a specific meaning (Morris and Hirst 1991). Consider as an illustration chain four from the example 2.4 above: The word "bank" is implicitly disambiguated here to the meaning of "a place to sit on". Another thing is that lexical chains provide a clue for the discourse structure and hence the overall meaning of a text. Consider Figure 2.1 as an illustration how lexical chains typically manifest themselves in larger texts. Here, the red chain runs through the whole text whereas the blue, yellow and green chains are only active in some paragraphs or sections of the text. The structure of the lexical chains mirrors the structure of the text.



When Dorothy was left alone she began to feel hungry. So she went to the cupboard and cut herself some bread, which she spread with butter. She set about making ready for the journey to the City of Emeralds. She took a little basket and filled it with bread from the cupboard. Then she looked down at her feet and noticed how old and worn her shoes were. At that moment Dorothy saw lying on the table the silver shoes that had belonged to the

Figure 2.1.: *How lexical chains typically manifest themselves in a normal text*[1]*. The left box shows an extract of the larger text in the right box. Words belonging to the same lexical chain are colored the same, e.g.* {feet, shoes, shoes} *forms one lexical chain, which is colored in red, and* {hungry, bread, butter, bread} *forms another lexical chain which is colored in green. On the right a red, blue, green and yellow chain can be seen. The red chain for example runs through the whole text whereas the others are only active in some parts of the text.*

Summarizing, lexical chains can be seen as the golden thread running through a text, where the words or expressions are grouped by a semantic relation.

[1] The text serving for illustrative purposes comes from "The Wonderful Wizard of Oz" by L. Frank Baum.

2.2 Lexical Chain Extraction Algorithms

Algorithms for extracting lexical chains mainly follow the generic framework presented by Barzilay and Elhadad (1997), which is shown in Listing 2.1.

Listing 2.1: *A generic framework for lexical chain extraction.*

```
 1 Input:   text
 2 Output: chainset
 3 /* initialize with an empty set of lexical chains */
 4 chainset $\leftarrow \emptyset$
 5 /* process each appropriate lexical item */
 6 for each candidate $\in$ candidates(text) do
 7     /* update the current chainset */
 8     chainset $\leftarrow$ update(chainset, candidate)
 9 end for
10 /* finalize the set of chains, e.g. delete temporary chains */
11 finalize (chainset)
12 return chainset
```

The purpose of the algorithm in Listing 2.1 is to deliver a set of lexical chains each comprising a subset of lexical items extracted from the given text. It starts with an empty chainset that is filled during the procedure. For each appropriate candidate item the current chainset is updated — either the item is inserted into an existing chain or new chains are generated or whatsoever. The behavior depends on the instantiation of the algorithm.

The key challenges of the generic algorithm are: *a*) the choice of adequate lexical items abstracted by *candidates(·)* (cf. Listing 2.1:6), and *b*) what needs to be done with the current item and chainset abstracted by *update(·)* (cf. Listing 2.1:8). These problems can be seen as parameters to be set by the concrete instantiation of the generic framework. In order to solve these problems additional computationally available knowledge about the vocabulary is needed.

Morris and Hirst (1991) used Roget's thesaurus (Roget 1852) as knowledge resource in their algorithm. In a *thesaurus* words are grouped into so-called categories which are again classified into classes and subclasses. The categories describe concepts and the inheritance hierarchy of classes describe certain relations between these abstract concepts. Categories are again divided into paragraphs that group closely related words of the same syntactic category which in turn are again separated by semicolons into smaller, finer grained groups. These subgroups may also contain pointers to other categories. The words themselves are indexed with categories and paragraphs which allows for the efficient retrieval of related words. Figure 2.2 shows the structure of Roget's thesaurus for an example category and the index for a word in that category.

- Class 1: Abstract Relations
  - I: · · ·
  - ⋮
  - VIII: Causation
    - ⋮
    - B: Cause And Effect
      - ⋮
      - 159: strength [Degree of power]
        1. NOUNS strength; power 157; energy 171; vigor, force; main force, physical force, brute force; spring, elasticity, tone, tension, tonicity. . . .
        - ⋮
      - ⋮
    - ⋮
  - ⋮
- ⋮

(a)

power	
authority	737.1
greatness	31.2
loudness	404.1
number	84.1
physical energy	171.1
power	157.1
strength	159.1
vigor	574.1

(b)

Figure 2.2.: *(a) The structure of Roget's thesaurus. (b) Index entry of the word* power *in a thesaurus. E.g. the word* power *in its sense of* strength *can be found in category* 159 *paragraph* 1.

Morris and Hirst (1991) defined five types of *thesaural relations* to be necessary for forming chains:

1. Two words share the same category pointer in their indexes.
2. In the index of one word exists a pointer to a category that contains a pointer to a category existent in the index of the other word.
3. A category in the index of one word is labeled with the other word, or the category contains the other word.
4. The indexes of the two words contain pointers to adjacent categories that are in the same (sub-)class, e.g. the first word index holds a pointer to category 159, and the second word index holds a pointer to category 160 or 158 that are in the same subclass.
5. The indexes of the two words contain a pointer to categories, each holding a pointer to the same category.

Morris and Hirst allow at most one transitive link, which means if word *a* is related to word *b* and word *b* is related to word *c*, then *a* is also considered to be related to *b*. Additionally they limited the search scope to at most three sentences back, which means a word is unrelated if it is more than three sentences back. Incidentally, Morris and Hirst defined the concept of *chain returns* in such a case, but in this thesis, this will not be further regarded.

When processing a candidate item one of the above conditions must hold between any of the items in existing chains and the candidate item, in which case, it is inserted into the chain which contains the related item; otherwise a new chain is created with the candidate item as single element. The candidate items are implicitly computed, by taking only those items into account, that are present in Roget's thesaurus — ignoring pronouns, prepositions, verbal auxiliaries and high frequency words such as *good*, *do*, etc. Although a machine readable thesaurus was not available to Morris and Hirst in 1991, they showed that the algorithm delivers the desired structure defined by lexical cohesion.

Even more common approaches like Hirst and St-Onge 1998; Barzilay and Elhadad 1997; Silber and McCoy 2002; Galley and McKeown 2003; Medelyan 2007, and many others, rely in their algorithms on WordNet® (Fellbaum 1998) as knowledge resource. WordNet is a lexical database for English, but variants with the same structure exist for a variety of languages. In WordNet, words of the same syntactic category are grouped into so-called *synsets*, which are sets of words that can be used synonymously, i.e. members of the same synset are interchangeable in a given context. Thus, words occurring in multiple synsets are considered to have multiple meanings. Synsets are assigned additional information, e.g. a gloss of the specific meaning of the synset, example sentences as a surrounding context, frequency counts, etc. Figure 2.3 shows exemplarily some synsets for a word in question.

Synsets in WordNet are linked by lexico-semantic relations within a specified scope. The scope of most relations is only within a certain syntactic category; links between categories exist, but are rare. A summarized list of relations in WordNet with a description and their scope can be found in Table 2.1.

The main difference between a thesaurus like Roget's and a lexical database like WordNet is the coverage of *systematic* and *non-systematic* relations. Systematic relations encode relations between words and phrases that are systematically classifiable like synonyms, antonyms, etc., while *non-systematic relations* are not systematically classifiable, e.g. collocations. A lexical database on one hand typically covers only systematic relations, whereas a thesaurus on the other hand covers both types of relations, systematic and non-systematic (Morris and Hirst 1991).

For illustration, consider the following example from Morris and Hirst, where the words in question are emphasized in italics:

(2.5) Mary spent three hours in the *garden*. She was *digging* potatoes.

power	
Noun	
power, powerfulness	*possession of controlling influence*
power	*(physics) the rate of doing work; measured in watts (= joules/second)*
ability, power	*possession of the qualities (especially mental qualities) required to do something or get something done*
power, force	*one possessing or exercising power or influence or authority*
exponent, power, index	*a mathematical notation indicating the number of times a quantity is multiplied by itself*
might, mightiness, power	*physical strength*
[. . .]	
Verb	
power	*supply the force or power for the functioning of*

Figure 2.3.: *Synsets in the* WordNet® *lexical database that contain the word* power.

Table 2.1.: *List of* WordNet® *relations between synsets. The scope is described by a shorthand notation of the syntactic categories nouns (N), verbs (V), adjectives (A), and adverbials (AV).*

Relation	Description	Scope
Hyperonymy / Hyponymy	synsets establish a kind-of relation, e.g. {*bed*} is a kind of {*furniture*}	N
Meronymy / Holonymy	synsets establish part-of relation, e.g. {*bedstead, bedframe*} is a part of {*bed*}	N
Troponymy	express a more specific manner of an event, e.g. {*move*} – {*jog*} – {*run*}	V
Entailment	synsets entailing another, e.g. {*buy*} – {*pay*}	V
Antonymy	opposing synsets, e.g. {*wet*} – {*dry*}	A
Similarity	semantically similar synsets (transitive antonymy), e.g. {*dry*} – {*parched, bone-dry, arid*}	A
Attributive	adjectives being attributes of nouns, e.g. {*heat (N)*} – {*hot (A)*}	N, A
Pertainymy	morphological related synsets sharing the same stem, e.g. {*observe (V)*} – {*observant (A)*} – {*observation (N)*}	N, V, A, AV

In WordNet the words “garden” and “digging” are not related, even not by a path of any length. Roget’s on the other hand contains a category “agriculture” which includes both words.

Hirst and St-Onge (1998) were one of the first who approached the lexical chaining task utilizing WordNet instead of a thesaurus where they defined the necessary *lexical relations* that replace the necessary *thesaural relations* originally defined by Morris and Hirst (1991). Hirst and St-Onge defined three types of relations for tying up words or phrases:

extra strong: holding between a word and its literal repetition;

strong: holding between words when they (*a*) are in the same synset, (*b*) one of their synsets are connected by antonymy or similarity, or (*c*) one word is a compound or phrase that contains the other and one of their synsets are connected by any kind of relation;

medium strong: holding between words when one of their synsets are connected by a path between two and five links of any kind of relation, where the path is restricted by the following rules: (*a*) if a hypernymic relation is in the path, it must be the first step, and (*b*) at most one generalization followed by a specification is allowed. Additionally a medium strong relation is assigned a weight calculated from the kind of relations used in the path and the length of the path. Details can be found in (Hirst and St-Onge 1998).

Hirst and St-Onge addressed the *candidates(·)* problem in Listing 2.1 by processing every word — ignoring pronouns, prepositions, verbal auxiliaries and high frequency words — and treating each word as a noun, i.e. for every appropriate word the noun file of WordNet was checked for existing synset entries, as the assumption is that most words of other grammatical categories that have a nominal form are likely to be semantically close to that form. Their choice of the “noun subnet” as entry point has some good reasons. First, it contains by far the most synsets and synset relations, and second, a part-of-speech tagger as a preprocessing step can be omitted.

While processing candidate items, Hirst and St-Onge keep a reference to the synsets in use in each of the chains. Based on the type of relation (extra strong, strong, medium strong) synsets are removed from further consideration. Unrelated candidate items — the first occurring candidate item also counts as unrelated — are inserted into a new chain, and the item keeps a reference to each of the synsets it occurs in. If a candidate item is related to two or more lexical items in distinct chains, the respective chains are merged. Next, if an extra strong relation is encountered, nothing is removed at all, since the words can not be disambiguated; if a strong relation is encountered, only the pairs of strongly connected synsets are kept; and if a medium strong relation is encountered only the highest weighted synset is kept.

Barzilay and Elhadad (1997) noted that by processing the candidate items in the order of their occurrence, lexical items may be disambiguated falsely. Barzilay and Elhadad illustrated the problem with an example, recited in example 2.6.

(2.6) Mr. Kenny is the person that invented an anesthetic machine which uses micro-computers to control the rate at which an anesthetic is pumped into the blood. Such machines are nothing new. But his device uses two micro-computers to achieve much closer monitoring of the pump feeding the anesthetic into the patient. [. . .]

According to Hirst and St-Onge's method, the underlined candidate words are processed in order of their occurrence. First, the word "Mr" is inserted into the chainset as a new chain, which is implicitly disambiguated, just because it is in only one synset. Second, the word "person" is inserted into the same chain because of a medium strong relation between "Mr" and "person", which keeps only a reference to the highest weighted synset named "a human being". Third, the word "machine" is processed. Because a strong relation holds between "person" and "machine" — in one of its synsets, "machine" is interpreted as a "very efficient person", which is in a hypernymic relation to "a human being" — it is inserted into the current {"Mr", "person"} chain. This is obviously the wrong behavior; the word "machine" is disambiguated falsely at this step. This tough problem of *word sense disambiguation* engages a lot of research in the field of computational linguistics.

Barzilay and Elhadad (1997) address this problem by keeping *a list of interpretations* of various *components* — a component is a set of relatable words, i.e. words that may but do not have to form a single chain, and an interpretation is the combination of words resulting in a specific set of chains — of the text, until in the end the strongest interpretation of each component is chosen. In order to discriminate the various interpretations qualitatively, Barzilay and Elhadad defined the strength for an interpretation to be the sum of the weights of the chains it inheres. The weight of a chain is the sum of the weights of existing relations in the respective chain. Barzilay and Elhadad defined two terms to be related by a lexical relation with a weight of (*a*) 10 if they are identical, or in the same synset, (*b*) 8 if one of their synsets are in a path in the hypernym hierarchy (*c*) 7 if one of their synsets are antonyms, (*d*) 4 if one of their synsets are meronyms, (*e*) 2 if one of their synsets are co-hyponyms with a maximum degree of 4. More details about the used weighting scheme can be found in Barzilay (1997, pp.32–33).

Barzilay and Elhadad's update procedure is best described with the example they provided. Reconsider therefore example 2.6 from above. In the second step, the word "person" is related to "Mr", so "person" is added to the component containing "Mr" and the interpretations are updated to:

interpretations before	interpretations after

{*Mr*}	{*Mr, person*}
	{*Mr*}, {*person*}

In the third step "machine" is related to "person" and "Mr", and so it is inserted into the component that contains "person" and "Mr". The interpretations are then updated to:

interpretations before	interpretations after

{*Mr, person*}	{*Mr, person, machine*}
	{*Mr, person*}, {*machine*}

{*Mr*}, {*person*}	{*Mr*}, {*person*}, {*machine*}
	{*Mr, machine*}, {*person*}
	{*Mr*}, {*person, machine*}

Next, “micro-computers” is inserted into the component containing “machine”, because of a relation between those two. Because of an exponential growth only the update of the first two interpretations resulting from the last step are shown:

interpretations before	interpretations after
{*Mr, person, machine*}	{*Mr, person, machine, micro-computers*} {*Mr, person, machine*}, {*micro-computers*}

{*Mr, person*}, {*machine*}	{*Mr, person*}, {*machine, micro-computers*}
	{*Mr, person*}, {*machine*}, {*micro-computers*}

[. . .]	

Proceeding until the last word was inserted results in a number of interpretations each having a certain weight. According to Barzilay and Elhadad the strongest interpretation and thus the resulting chainset for the running example is {{“Mr”, “person”}, {“machine”, “micro-computers”, “machines”, “device”, “micro-computers”, “pump”}}, which clearly follows the intuition. In order to limit the computational complexity, the search scope is limited to the segment, where the candidate item is located. Segments are computed beforehand using the segmentation algorithm by Hearst (1994).

Barzilay and Elhadad (1997) consider nouns as possible candidate items — excluding modifiers of noun compounds. Nouns are identified by a part-of-speech tagger and compound nouns are detected heuristically. Unfortunately the algorithm has an exponential complexity in time and space — exclusive of the three preprocessing steps — which makes it impractical for the use with larger documents.

Silber and McCoy (2002) subsequently introduced an efficient variant of Barzilay and Elhadad’s algorithm that has an overall complexity of $O(n)$, where n is the number of nouns used in the document. Silber and McCoy’s algorithm follows mainly Barzilay and Elhadad’s methodology, but instead of limiting the search scope to segments, they use different weights of lexical relations dependent on the distance between the words and the kind of relation. A summary of their weights is shown in Table 2.2.

For the reasons of efficiency, Silber and McCoy recompiled the WordNet noun database beforehand, so that it can be efficiently accessed as a large array in memory. Furthermore, they matched WordNet sense numbers to the array indexes in order to use a data structure that implicitly stores every

Table 2.2.: *Silber and McCoy's (2002) lexical relation factors.*

	Distance between the two words			
lexical relation	≤ 1 sentence	≤ 3 Sentences	same Paragraph	other
Repetition	1	1	1	1
Synonymy	1	1	1	1
Hypernymy	1	0.5	0.5	0.5
Co-Hyponymy	1	0.3	0.2	0

interpretation without actually creating it. With this technique, smaller documents are processed about one hundred times faster than with Barzilay and Elhadad's technique; larger documents are actually processable now at all.

Galley and McKeown (2003) mentioned nevertheless that, although Barzilay and Elhadad's methodology better approaches the word sense disambiguation (WSD) problem than other methods before, it still lacks accuracy. Currently, WSD is implicitly resolved when chains are selected. Galley and McKeown suggest separating WSD from the actual chaining process, by selecting first the correct sense for a word; the lexical chains then implicitly become apparent.

Technically speaking, they introduced a so-called *disambiguation graph*, which is just another convenient view of an interpretation. In such a graph, a word is represented as a node split into its various synset senses in WordNet. An edge represents the weighted lexical semantic relation between two words and their synsets. An illustration of a node in a disambiguation graph is shown in Figure 2.4.

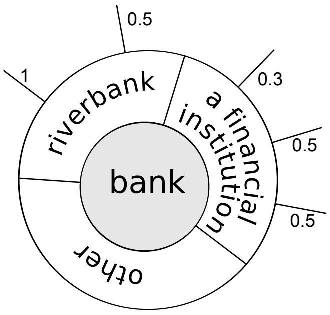


Figure 2.4.: *The word* "bank" *in an illustrative disambiguation graph. Here* "bank" *has some weighted lexical semantic relations via the senses* "riverbank" *and* "financial institution". "Other" *is an omissible placeholder for synsets that did not form any relations. In this case the word* "bank" *would be disambiguated to* "riverbank", *because the sum of its weights is highest.*

The first step of the algorithm is to create the disambiguation graph. Galley and McKeown (2003) confine their method to occurrences of words instead of inserting each word instance. In other words, if a word "bank" occurs ten times in a text, then a node "bank", including the found relations, is created when it is first encountered, and the other nine occurrences just update the relations of that node. Thus, a "repetition" relation does not occur explicitly in the graph. The weights of lexical relations, empirically evaluated by Galley and McKeown, are summarized in Table 2.3.

Table 2.3.: *Galley and McKeown's (2002) lexical relation factors.*

	Distance between the two words			
lexical relation	≤ 1 sentence	≤ 3 Sentences	same Paragraph	other
Synonymy	1	1	0.5	0.5
Hypernymy	1	0.5	0.3	0.3
Co-Hyponymy	1	0.3	0.2	0

In the second step of the algorithm all nodes of the graph are disambiguated by choosing that sense for a node, whose sum of edge weights is maximal, and hence, edges of other senses are removed. By doing so Galley and McKeown follow the *one sense per discourse* assumption of Gale et al. (1992), who discovered that in a normal well-written text all occurrences of a certain word share, with 98% chance, the same sense. The third step is then the conversion of the pruned graph to lexical chains by inserting all occurrences of words that are connected in the resulting disambiguated graph, into the same chain. According to Galley and McKeown, the results of a WSD task, that uses lexical chains, have remarkably improved when using their lexical chains versus the lexical chains computed by Silber and McCoy (2002) or Barzilay and Elhadad (1997).

Medelyan (2007) approached the lexical chaining task also in a way based on graph theory, but instead of explicitly disambiguating the nodes she applied a *graph clustering* algorithm, and mapped the resulting node clusters back to chains, thus disambiguating the words implicitly.

Medelyan's idea is that lexical chains can be mapped to a graph in which nodes represent words, edges represent lexical semantic relations between words and the strength of the cohesiveness of the document can be measured in terms of the *diameter* of that graph. The *graph diameter* is defined as the maximum of the minimum *distances* between every pair of nodes. In case of Medelyan's approach the graph is undirected and unweighted, which means that the minimum distance to get from node v_i to node v_j is the minimum number of hops, i.e. the number of edges that have to be traversed.

Let $G = (V, E)$ be a graph representing a lexical chain with V being the set of nodes, E being the set of edges and m being the diameter of G. The maximum value m can take is defined as $m_{max} = |V| - 1$. Based on the graph diameter, Medelyan defines three types of lexical chains: (a) *strongly cohesive chains* with m $= 1$, which implies that the graph is fully connected and each word is directly related to each other, (b) *moderately cohesive chains*: with $1 < m < m_{max}$, and (c) *weakly cohesive chains*: with $m = m_{max}$.

The first step of Medelyan's algorithm is to build (weak) lexical chains, where every n-gram that occurs in WordNet is considered a candidate item. Next, the chainset is updated analog to Hirst and St-Onge's strategy. Unfortunately Medelyan does mention neither the kind of relation nor the search scope for related items, so we need to assume that a candidate item is related to every other item if any kind of lexical relation between these two holds. Finalizing, all those lexical chains, that have a diameter of $m > 3$, are considered to be weakly cohesive and thus are processed by a *graph clustering* algorithm in order to get a set of lexical chains each being cohesively stronger than the one chain before. *Graph clustering algorithms* provide a class label for each node and the main goal is to assign the same label for a group of nodes that show a strong affinity. Specifically, Medelyan used the *ChinesWhispers* graph clustering algorithm (Biemann 2006), which will be discussed in detail in Section 5.2.2. Finally, nodes, respectively words that are assigned the same label are grouped into the same lexical chain. Figure 2.5 illustrates a possible result after chaining and clustering example 2.6. Medelyan reported that lexical chains produced by this technique delivered again better results in summarization and in keyphrase indexing.

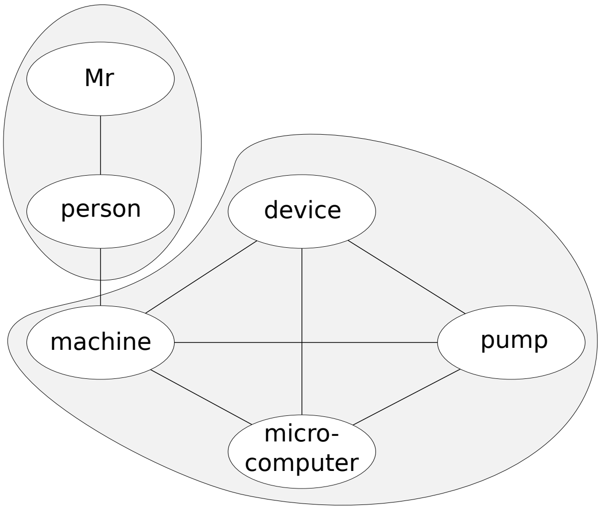


Figure 2.5.: *Graph clustering example. Consider the whole graph to be a moderately connected chain with a diameter $m = 3$. After applying graph clustering the two subgraphs represent two distinct strongly cohesive lexical chains, each with a diameter of $m = 1$.*

2.3 Desiderata for Lexical Chain Annotation

Other algorithms beside the ones explained before exist, each extracting lexical chains in a slightly different way and with another goal in mind. E.g. Okumura and Honda (1994) performed word sense disambiguation utilizing lexical chains, Stairmand (1996) used lexical chains in information retrieval, Green (1996) used lexical chains for creating hypertext links, Al-Halimi and Kazman (1998) indexed video conferences in a database, Moldovan and Novischi (2002) approached the task of question answering, Jarmasz (2003) introduced the ELKB[2] and used it for lexical chain extraction, Stokes et al. (2004) segmented news story with the help of lexical chains, Teich and Fankhauser (2005) analyzed lexical chains with respect to different registers, Reeve et al. (2006) used lexical chains for summarization in the biomedical domain, Yang and Powers (2006) utilized the EAT[3] for extracting lexical chains and performed word sense disambiguation, Ercan and Cicekli (2007) extracted keywords using lexical chains, Marathe and Hirst (2010) utilized DMCDs[4] for extracting lexical chains and evaluated them by text segmentation, etc. The list seems endless, and in fact, even when they are not extracted as an intermediate tool, lexical chains are implicitly assumed in nearly every natural language processing task that deals with some kind of text understanding simply because it is assumed that the text is cohesive.

The main problem when developing a new lexical chaining algorithm is the comparison to previous systems. Most of the techniques mentioned before claim to produce better lexical chains than the methods before. This is substantiated by the evaluation of a certain task like summarization, word sense disambiguation, or keyphrase extraction using their computed lexical chains and lexical chains produced by other systems. Since lexical chains are mostly used as an intermediate step and the parameter details are very fine grained, little changes may have a deep impact in the outcome of the certain task. Also, reproducibility is mostly impossible due to the brevity in given details. Thus the result of each evaluation is, in part, always subjectively influenced. What is generally needed are objective criteria for directly comparing lexical chains.

Strictly speaking a generally accepted corpus is needed, manually annotated with lexical chain information, available for everybody, so that every new algorithm may be evaluated using this corpus, presenting its result using a generally accepted evaluation measure. Both claims will be addressed in the next chapters, starting with the question for a good lexical chain comparison measure.

[2] http://rogets.site.uottawa.ca – The Electronic Lexical Knowledge Base (ELKB) is an electronic version of Roget's thesaurus.

[3] http://www.eat.rl.ac.uk – The Edinburgh Associative Thesaurus (EAT) contains spontaneous responses for a given word.

[4] Distributional measures of Concept Distance (DMCDs) combine distributional co-occurrence information with semantic information from a lexicographic resource (Marathe and Hirst 2010).

3 Comparing Lexical Chains

Once different sets of lexical chains of the same text are available — coming either from manual annotation or as a result of automatic chaining algorithms — they need to be compared in order to qualitatively say how similar or how different they are; i.e. the degree of agreement must be measured. Unfortunately, the comparison of lexical chains is a non-trivial task. A first methodology was performed by Morris and Hirst (2004) in a user study where annotators were asked to (a) identify "word groups" (i.e. lexical chains) in various texts, and (b) mark words that they perceived to have a direct relation. The resulting data was then analyzed as follows: First, Morris and Hirst measured the pairwise agreement between annotators on the lexical items they used in terms of accuracy and then averaged it with the accuracy of all possible pairs of annotators. Hereby, they collected all items annotated by a pair of annotators and used the complete set of items as the reference set for computing the relative amount of items on which the annotators agreed. The results of all pairs of annotators are then averaged:

$$\textit{Item-Accuracy} = \frac{1}{n(n-1)/2} \sum_{i,j \,\in Annotators} \textit{Item-Accuracy}_{ij} \qquad \text{where } \; n = |Annotators| \tag{3.1}$$

$$\textit{Item-Accuracy}_{ij} = \frac{\#\ items\ annotator_i\ AND\ annotator_j\ identified}{\#\ items\ annotator_i\ OR\ annotator_j\ identified} \tag{3.2}$$

In the next step, they measured the agreement on prevalent word pairs that were often marked as directly related within the word groups. Here, Morris and Hirst matched the word groups of individual annotators to global ones (i.e. word groups annotated by the majority of annotators), and used only those word pairs that were marked by at least 50% of the annotators. On that basis, they computed the average accuracy for each word pair as:

$$\textit{Word-Pair-Accuracy} = \frac{1}{n} \sum_{i \,\in Annotators} \textit{Word-Pair-Accuracy}_i \tag{3.3}$$

$$\textit{Word-Pair-Accuracy}_i = \frac{\#\ word\ pairs\ marked\ by\ annotator_i}{\#\ word\ pairs\ marked\ any\ annotator} \tag{3.4}$$

Although the intention of Morris and Hirst for using these measures is to qualitatively measure the subjectivity of lexical chaining, the methodology they introduced can also be used to compare lexical chains produced by different annotators. However, the manual selection steps are not acceptable in an automatic evaluation framework.

Another approach for measuring the similarity of lexical chains was performed by Hollingsworth and Teufel (2005). They treated each individual lexical chain as a set of words and all chains of a particular document are treated as a set of lexical chains, a so-called *chain set*. They compared four measures in terms of their suitability for measuring similarity between chain sets produced by different annotators:

1. *cosine similarity metric*
2. *Kullback Leibler distance*
3. *strict term overlap*
4. *partial term overlap* (analogous to *strict term overlap*, but splitting compound terms into their individual parts before measuring term overlap).

More details on these can be found in Hollingsworth and Teufel (2005). However, all these measures are constrained to only compute the similarity between two individual chains and not between two complete chain sets. To overcome this, Hollingsworth and Teufel averaged the results of the computation of each chain in chain set A to the best matching chain in chain set B:

$$\textit{chain-set-similarity}(A,B) = \frac{1}{n}\sum_{x\in A}\max_{y\in B}(sim(x,y))\,|x| \qquad \text{with } \; n = \sum_{x\in A}|x|\,, \tag{3.5}$$

where $sim(x,y)$ is one of the four measures above.

In their investigation Hollingsworth and Teufel noted that the two *term overlap* scores matched the intuitive behavior more than the other two measures and the *partial term overlap* in particular creates more intuitive results than the *strict term overlap*, but still does not capture all desired cases.

Nelken and Shieber (2007) extended the idea of using sets for comparison by introducing the concept of *clusterings*. Within this approach a lexical chain is interpreted as a *cluster* of words and the sum of lexical chains in a document are interpreted as a *clustering* (i.e. a set of *clusters*). A detailed definition of clustering is found in Section 3.1. Nelken and Shieber measured the similarity between different clusterings in terms of *purity* and *entropy*. Let n be the total number of words and let $C = \{c_1, \ldots, c_k\}$ and $C' = \{c'_1, \ldots, c'_{k'}\}$ be two clusterings of two annotators. Further, let

$p_{ij} = |c_i \cap c'_j| \,/\, |c_i|$ be the fraction of words in the clusters c_i and c'_j with respect to c_i. The *purity* and *entropy* of C is then computed as:

$$purity(C) = \frac{1}{n}\sum_{i=1}^{k} purity(c_i)|c_i| \qquad purity(c_i) = \max(p_{ij}) \tag{3.6}$$

$$entropy(C) = \frac{1}{n}\sum_{i=1}^{k} H(c_i)|c_i| \qquad H(c_i) = -\sum_{j=1}^{k'} p_{ij}\log p_{ij}\ . \tag{3.7}$$

In their investigation, Nelken and Shieber observed that the *entropy* measure opposes the intuitive behavior if too many false assignments occur. In such a case the value p_{ij} is often zero, which leads $p_{ij}\log p_{ij}$ to be zero per definition, which then leads to a smaller entropy value. This also happens if too many chains occur. Entropy can be seen as a measure of *unorderedness* and a smaller value indicates that the clustering is ordered, and this contradicts the intuition how a measure should behave when comparing lexical chains.

In general, the treatment of lexical chains as sets of words (Hollingsworth and Teufel 2005) or as clusters of words (Nelken and Shieber 2007) has two major drawbacks: First, repetitions of individual words in the same lexical chain are ignored, and thus a false assignment of a single occurrence of a word drastically weights down the similarity score. Second, it also ignores the fact that same words with different meaning (homonyms and polysemes) are correctly located in different chains, but will be punished for this by the similarity measures. Consider the examples below with some lexical chains (left) and their respective set or cluster representation as defined by Hollingsworth and Teufel (2005) and Nelken and Shieber (2007) (right).

(1) { *money, bank, money, money, bank, money, bank* } ⟶ { *money, bank* }

(2) { *money, bank* } ⟶ { *money, bank* }

(3) { *boat, river, bank* } ⟶ { *boat, river, bank* }

In the first chain, it is easy to see, that some important information is lost, namely the repetitions of the words *bank* and *money*. Thus, comparing the first chain with the second chain using their set representation leads to perfect similarity. In the lexical chains 2 and 3 the word *bank* is located in both chains, but assuming the lexical chains are correct, the mention of the word *bank* in chain 3 refers to a different meaning than the word *bank* in chain 2 — think of *bank* as a financial institution or as a slope in a river where boats can crash. The similarity measure would reveal similarity between the two chains although they are clearly different.

This thesis derives a methodology which is also based on the idea of interpreting lexical chains as clusters, but in contrast to the suggestion of Hollingsworth and Teufel (2005) or Nelken and Shieber (2007), the elements in a cluster will be some unique identifier of the respective word in the text. This way, repetitions of certain words will not be truncated. Additionally an extensive investigation will be presented, which illustrates the suitability of various standard measures used in clustering comparison, for the task of comparing lexical chains.

3.1 Clustering Defintion

A clustering C is a partition of a non-empty dataset D into K non-empty sets $c_1, \ldots, c_K$ called clusters such that

$$N = |D| \qquad N \geq 1 \tag{3.8}$$

$$D = \bigcup_{k=1}^{K} c_k \tag{3.9}$$

$$\emptyset = c_i \cap c_j \qquad i, j \in \{1, 2, ..., K\} \wedge i \neq j \tag{3.10}$$

$$\emptyset \neq c_i \qquad i \in \{1, 2, ..., K\} \,. \tag{3.11}$$

In words: the dataset D must consist of at least one element and each element in D must be existent in exactly one of the clusters $c_1, \ldots, c_K$ and all clusters c_k contain at least one element. Thus, the number of maximally usable clusters is upper bounded by the number of elements in D and lower bounded by one

$$1 \leq K \leq N \,. \tag{3.12}$$

3.2 Lexical Chains as Clusters

Throughout this thesis, a clustering is defined to contain all lexical chains of a particular document. The elements of the dataset D are defined to be unique identifiers of the words used in a lexical chain. D thus contains only the words that are used in any of the chains that were identified in the document. The unique identifier of a word is defined as the position of a word in the source document. Though, other identifiers are possible. Consider the following illustrative example:

(3.1) When Dorothy was left alone she began to feel hungry. So she went to the cupboard and cut herself some bread, which she spread with butter. She set about making ready for the journey to the City of Emeralds. She took a little basket and filled it with bread from the cupboard. Then she looked down at her feet and noticed how old and worn her shoes were. At that moment Dorothy saw lying on the table the silver shoes that had belonged to the Witch of the East. She took off her old leather shoes and tried on the silver ones, which fitted her as well as if they had been made for her. Finally she picked up her basket. She closed the door, locked it, and put the key carefully in the pocket of her dress. And so, with Toto trotting along soberly behind her, she started on her journey.

some lexical chains from the example text 3.1:

(1) { *hungry, bread, butter, bread* }

(2) { *feet, shoes, shoes, shoes* }

conversion into clustering representation:

$$D = \{\ hungry_{10},\ bread_{21},\ butter_{26},\ bread_{49},\ feet_{59},\ shoes_{67},\ shoes_{80},\ shoes_{96}\ \}$$

$$C = \{\ \{\ hungry_{10},\ bread_{21},\ butter_{26},\ bread_{49}\ \},\ \{\ feet_{59},\ shoes_{67},\ shoes_{80},\ shoes_{96}\ \}\ \}$$

$$\Longrightarrow K = 2, \quad N = 8$$

Mind that the lexical chains only serve for illustrative purposes and are not necessarily complete. All the above definitions of a clustering (Eq. (3.8)–(3.11)) are implicitly satisfied by the definition of lexical chains and some trivial assumptions:

eq. (3.8) trivial, only documents with at least one lexical chain are considered

eq. (3.9) trivial, all words that are chained are part of the same document

eq. (3.10) by definition, a word may only be part of exactly one or none lexical chain

eq. (3.11) by definition, a lexical chain must consist of at least one member

In the following we will write cluster when we refer to the cluster representation of lexical chains. Further we will use symbolic notation for the elements in clusters.

3.3 Comparing Clusterings

Suppose a clustering C was built of lexical chains in a particular document and suppose further a second clustering C' was built of another set of lexical chains from the same document. The task is now to measure how close C and C' are.

3.3.1 Cluster Conformation

Though created from the same document, the choice of lexical items during the chaining process is accompanied by subjectivity and thus the datasets D and D' spanned by C and C' respectively may be different. In order to compare a clustering C to a clustering C' it is a convenient preprocessing step to produce conformity within the underlying datasets such that $D = D'$. Since we do not want to ignore any decisions made in any of the clusterings, we transform C and C' to reflect the union of the datasets of each other ($D \cup D'$).

Two options are now possible conformations: (a) inserting each missing item as a single cluster, or (b) inserting each missing item into the same single cluster. Because each lexical item is assigned a certain meaning, mixing meanings, as would be the case in option b, is counterintuitive, and thus option a is chosen, and missing items are inserted as single clusters as formalized below:

$$\begin{aligned} C &:= C \cup \{d'_1\} \cup \cdots \cup \{d'_n\} \qquad & d'_1, \ldots, d'_n &\in D' \setminus D \\ C' &:= C' \cup \{d_1\} \cup \cdots \cup \{d_n\} \qquad & d_1, \ldots, d_n &\in D \setminus D' \,. \end{aligned} \tag{3.13}$$

As an illustrative example consider $C = \{\{a, b\}, \{c\}\}$ and $C' = \{\{a, d\}, \{e, f\}\}$. After applying the equations in (3.13) we have $C = \{\{a, b\}, \{c\}, \{d\}, \{e\}, \{f\}\}$ and $C' = \{\{a, d\}, \{e, f\}, \{b\}, \{c\}\}$. Note that conformity is a prerequisite of the measures below, it is thus assumed that two clusterings C and C', that are to be compared, are conformed by the above technique before measuring similarity. We then only speak of a dataset D instead of D and D' because $D = D'$, and C and C' are simply two different clusterings of the same dataset.

3.3.2 Desirable Properties

As stated by Meilă (2005) and Amigó et al. (2009) a best clustering comparison measure for the general case does not exist. Meilă (2005) proved that it is simply impossible for any clustering comparison measure to satisfy all comparison criteria she presented. As it is, she examined various measures and proposed one — the *variation of information* (Meilă 2003, 2007) — that fulfills all her criteria but one. Amigó et al. (2009) did a similar investigation and compiled some formal constraints that any clustering measure should satisfy. In their work, they pointed out that the only clustering measure satisfying all of their four constraints is the B^3 measure by Bagga and Baldwin (1998). However, Meilă as well as Amigó et al. stressed that the clustering measure to use highly depends on the task at hand.

In the following some basic properties are listed that should be reflected by a good measure for comparing lexical chains:

1. The measure should be a value between zero and one:

$$sim(C,C') \in [0,1] \subset \mathbb{R} .$$

2. The measure should be maximal when $C = C'$ and not maximal when $C \neq C'$:

$$sim(C,C') = \begin{cases} 1 & \text{if } C = C' , \\ < 1 & \text{otherwise} \end{cases} .$$

   Note that a distance measure $d(C,C')$ with $0 \leq d(C,C') \leq 1$ and $d(C,C') = 0$ if $C = C'$ can also be interpreted as a dissimilarity measure and thus converted to a similarity measure:

$$sim(C,C') = 1 - d(C,C') .$$

3. The measure should be minimal when D and D' have no items in common before conformation (cf. Sec. 3.3.1):

$$sim(C,C') = 0 \quad \text{if} \quad D \cap D' = \emptyset .$$

   After applying the equations in (3.13) D and D' share the same elements per definition.

4. The measure should be symmetric:

$$sim(C,C') = sim(C',C) .$$

   This is due to the fact that the measure will be used for inter-annotator agreement, where there is no way to distinguish which clustering is the gold clustering. Note that symmetry can always be forced by $sim_{symmetric}(C,C') = 1/2\, sim_{asymmetric}(C,C') + 1/2\, sim_{asymmetric}(C',C)$.

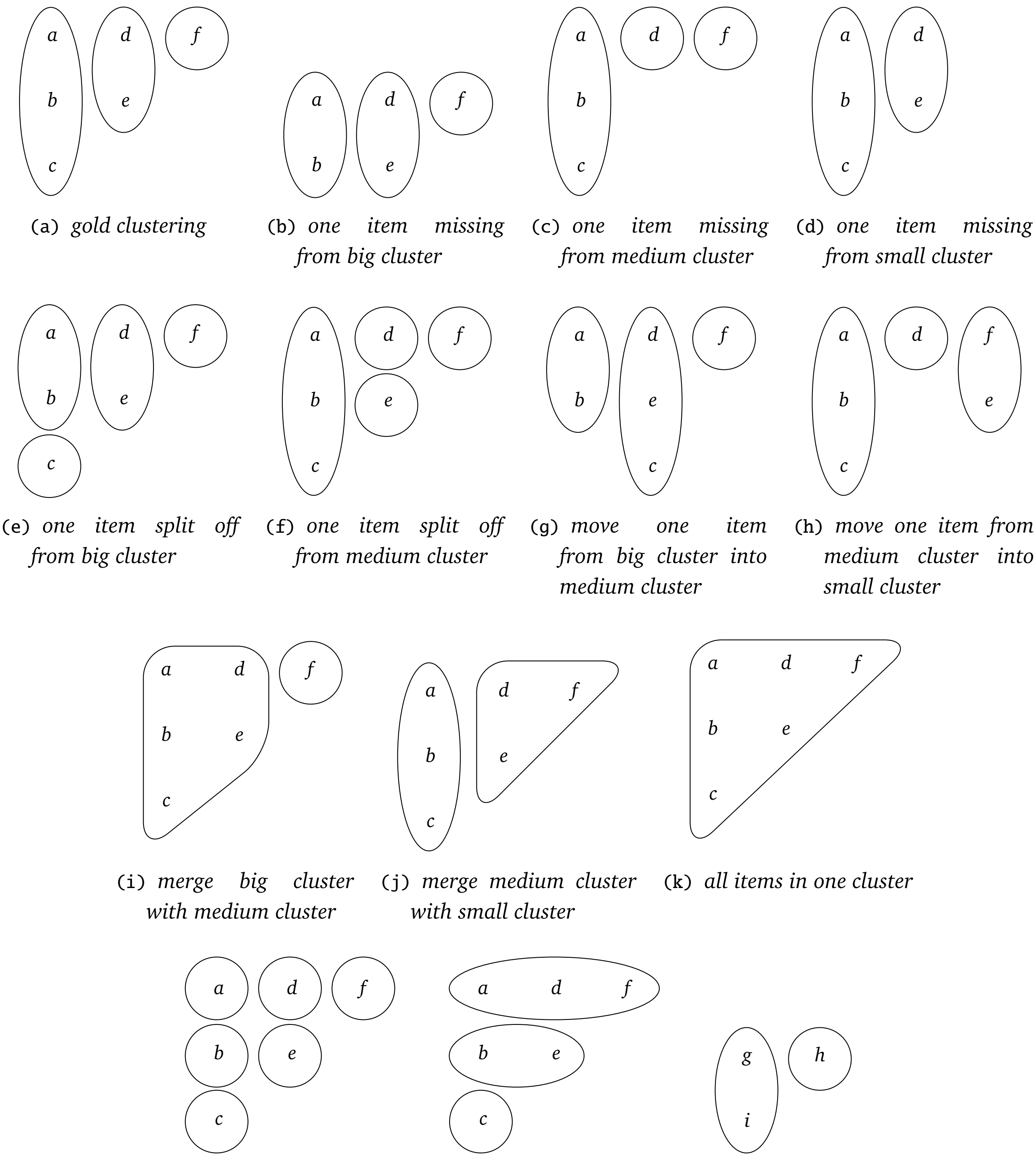
5. Simple basic transformations should be reflected by the same difference in the value:

$$|sim(C, trans_1(C)) - sim(C, trans_2(C))| = 0 ,$$

   where $trans_1(\cdot)$ and $trans_2(\cdot)$ are different basic operations, for example splitting a cluster into two or merging two clusters into one, etc. Figure 3.1b to 3.1j show some clusterings where just some basic operations are needed to get the gold clustering in 3.1a.

The list of properties is not necessarily complete, other properties, e.g. the independence of dataset size or the number of clusters, are also valuable, but the above are perceived as the most important ones for the task of comparing two separate sets of lexical chains extracted from the same document.

In the next sections some selected measures will be analyzed for their suitability of the current task. For this, the measures will be i.a. used for comparing artificially created clusterings as illustrated in Figure 3.1 where the desired clustering is shown in 3.1a and the similarity is calculated against the clusterings in 3.1b–3.1n.



(a) *gold clustering*

(b) *one item missing from big cluster*

(c) *one item missing from medium cluster*

(d) *one item missing from small cluster*

(e) *one item split off from big cluster*

(f) *one item split off from medium cluster*

(g) *move one item from big cluster into medium cluster*

(h) *move one item from medium cluster into small cluster*

(i) *merge big cluster with medium cluster*

(j) *merge medium cluster with small cluster*

(k) *all items in one cluster*

(l) *each item in one cluster*

(m) *orthogonal clustering*

(n) *no items in common*

Figure 3.1.: *Various clusterings resulting from transformations of the clustering in (a). Descriptions of the individual clusterings refer to the* gold clustering *in (a). Note that all clusterings can be expressed as the opposite when switching the roles of the* gold clustering *with the* system clustering*. E.g. (b)–(d) can be interpreted as* one item extra in big/medium/small cluster *when switching the role of the respective clusterings with the gold clustering.*

3.4 Clustering Comparison Measures

Clustering comparison measures can be split into four main categories (Amigó et al. 2009): (*a*) set based comparison measures (Sec. 3.4.2) including the *closest cluster* F_1- and the *K-measure* (*b*) pairwise element comparison measures (Sec. 3.4.3) including the *pairwise* F_1*-measure* and the adjusted Rand index (*c*) information theory based comparison measures (Sec. 3.4.4) including the *variation of information* and some normalized variants, the *V-measure* and the *normalized mutual information* (*d*) single element based comparison measures, also called the B^3 family (Sec. 3.4.5) including the B^3 *precision-*, *recall-*, and F_1*-measure*, and (*e*) measures based on edit distances (Sec. 3.4.6) including the *basic merge distance* and the *normalized basic merge distance*.

3.4.1 Contingency table

A *contingency table* is a handy tool for comparing two different clusterings C and C' from the same dataset D in terms of the items the individual clusters share. In information retrieval, a contingency table is also referred to as *confusion matrix*. Equation (3.14) shows how a contingency table is structured.

$$\begin{array}{c|cccc|c}
 & c'_1 & c'_2 & \dots & c'_{K'} & \sum \\
\hline
c_1 & n_{11} & n_{12} & \dots & n_{1K'} & n_{1.} \\
c_2 & n_{21} & n_{22} & \dots & n_{2K'} & n_{2.} \\
\vdots & \vdots & \vdots & & \vdots & \vdots \\
c_K & n_{K1} & n_{K2} & \dots & n_{KK'} & n_{K.} \\
\hline
\sum & n_{.1} & n_{.2} & \dots & n_{.K'} & N
\end{array} \tag{3.14}$$

$$n_{ij} = |c_i \cap c'_j|, \quad n_{i.} = |c_i|, \quad n_{.j} = |c'_j|$$

In such a contingency table, the entry n_{ij} is simply the number of items that are in the clusters c_i and c'_j, where $c_i \in C$, and $c'_j \in C'$. Let N be the number of items in D. Further let K be the number of clusters in C and K' the number of clusters in C' respectively.

3.4.2 Set based comparison

Set based comparison measures evaluate for each cluster in a clustering the best matching cluster in another clustering.

Closest Cluster F$_1$

As defined in (Benjelloun et al. 2009; Menestrina et al. 2010), the closest cluster F$_1$ measure (called ccF_1 henceforth) finds for a cluster c_i in C the "closest" cluster c'_j in C' based on the jaccard coefficient

$$J(c_i, c'_j) = J(c'_j, c_i) = \frac{|c_i \cap c'_j|}{|c_i \cup c'_j|} = \frac{n_{ij}}{|c_i \cup c'_j|}\,, \tag{3.15}$$

and computes the closest cluster precision (ccP) and recall (ccR) values according to

$$ccP(C, C') = \frac{1}{K'} \sum_j \max_{c_i \in C} J(c_i, c'_j) \tag{3.16}$$

$$ccR(C, C') = \frac{1}{K} \sum_i \max_{c'_j \in C'} J(c_i, c'_j)\,. \tag{3.17}$$

The F$_1$ measure is then defined as the harmonic mean between precision and recall

$$ccF_1(C, C') = \frac{2 \times ccP(C, C') \times ccR(C, C')}{ccP(C, C') + ccR(C, C')}\,. \tag{3.18}$$

Figure 3.2a shows how the measure interacts when comparing the gold clustering in 3.1a with the other clusterings in Figure 3.1. Simple transformations are almost equally valued, which is indicated by an almost equally height in b – j. Further it is able to recognize the missing item in d. However, an orthogonal clustering as in 3.1m is not recognized.

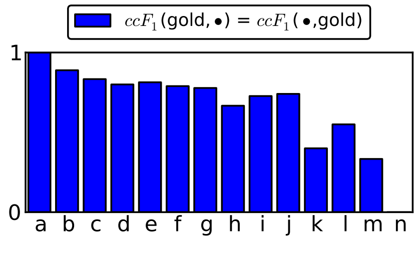
K

The K measure as defined in (Ajmera et al. 2002) sums the similarity values of all cluster pairs, and computes the geometric mean of the average cluster purities in both directions ($acp(C, C')$ and $acp(C', C)$):

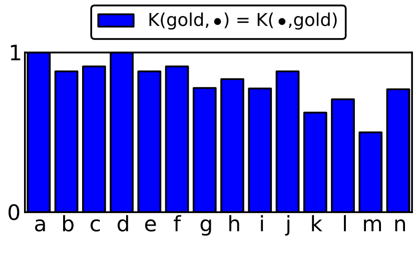
$$acp(C, C') = \frac{1}{N} \sum_i \frac{1}{n_{i.}} \sum_j n_{ij}^2\,, \qquad acp(C', C) = \frac{1}{N} \sum_j \frac{1}{n_{.j}} \sum_i n_{ij}^2 \tag{3.19}$$

$$K(C, C') = \sqrt{acp(C, C') \times acp(C', C)}\,. \tag{3.20}$$

Figure 3.2b shows how the measure interacts with the synthetic data from Figure 3.1. The major problem of this measure is that it does not recognize missing or additional items (cf. d), and because of this, it does not penalize clusterings according to property 3. As a result of this, the clustering in n, which has no elements in common with the clustering in a, is considered to be a better clustering than the clustering in k or l, which have all the elements of a. Obviously, this problem is a result of the conformation step in Section 3.3.1. However, this is an undesired behavior.



(a)



(b)

Figure 3.2.: *Set based comparison measure results of the synthetic clusterings evaluation. (a) shows the results from the* closest cluster F_1 *and (b) shows the results of the K measure.*

3.4.3 Pairwise element comparison

Another class of clustering comparison measures is that of the *pairwise comparison measures* which consider all pairs of items that are located in the same cluster.

Pairwise F_1

The pairwise F_1 measure (pF_1) is probably the most widely used evaluation measure given in almost every clustering evaluation scenario. According to the methodology in (Manning et al. 2008), the pF_1 is computed as follows: First a function $pairs(\cdot)$ is needed, which takes as input a clustering, and produces as output all unique combinations of items that are in the same cluster. Consider the following example:

$$C = \{\{a,b,c\},\{d,e\}\}$$
$$pairs(C) = \{<a,b>,<a,c>,<b,c>,<d,e>\}\ .$$

We can then count these pairs and create a view of correct and false decisions that have been made to get from C to C'. The *true positives* (TP) as the number of pairs in C and C' and the *true negatives* as the number of pairs neither in C nor in C', can be seen as correct decisions, whereas the *false negatives* (FN) and the *false positives* (FP) can be seen as the two types of errors that can occur. First the FN is the number of pairs that are present in C but not in C', and second, the FP is the number of pairs that are present in C' but not in C. The formulae for these are:

$$\begin{aligned} TP &= |pairs(C) \cap pairs(C')| \qquad & FN &= |pairs(C) \setminus pairs(C')| \\ FP &= |pairs(C') \setminus pairs(C)| \qquad & TN &= \binom{N}{2} - TP - FN - FP \end{aligned} \tag{3.21}$$

An alternative computation of these values is by means of combinatorics. Using a contingency table, the TP-, TN-, FP-, and FN-values can be computed as:

$$\begin{aligned} TP &= \sum_{i,j} \binom{n_{ij}}{2} & FN &= \sum_{i} \binom{n_{i.}}{2} - TP \\ FP &= \sum_{j} \binom{n_{.j}}{2} - TP & TN &= \binom{N}{2} - TP - FN - FP \end{aligned} \tag{3.22}$$

Finally, pairwise precision (pP), pairwise recall (pR) and pairwise F$_1$ (pF_1) is computed as:

$$pP(C,C') = \frac{TP}{TP+FP} \tag{3.23}$$

$$pR(C,C') = \frac{TP}{TP+FN} \tag{3.24}$$

$$pF_1(C,C') = \frac{2 \times pP(C,C') \times pR(C,C')}{pP(C,C') + pR(C,C')} \tag{3.25}$$

Figure 3.3a shows the results of the synthetic test. The strength of the pF_1 measure is that it successfully recognizes and punishes the orthogonal clustering (cf. m) and different datasets (cf. n). The major drawback of pairwise comparison measures can be observed in l, where each item is in a single cluster. Since the computation is based on pairs, it is a natural assumption that clusters should consist of at least two items. This is assumption does not cohere with the intuitive lexical chain comparison, because here, single element chains are possible.

Adjusted Rand Index

The *adjusted Rand index* (Hubert and Arabie 1985) is another measure from the domain of pair comparison measures. It is an advanced version of the *Rand index* (Rand 1971), which is simply the *accuracy* of the correct pairs versus all pairs:

$$RI(C,C') = \frac{TP+TN}{TP+TN+FP+FN} \ . \tag{3.26}$$

Hubert and Arabie developed the *adjusted Rand index* (*ARI*) as a chance corrected version of the *Rand index*, by introducing the terms *index*, *expected index*, and *maximum index*. The full derivation of these can be reviewed in (Hubert and Arabie 1985). The *ARI* measure is computed as:

$$index = TP \tag{3.27}$$

$$expected\ index = \frac{(TP + FP) \times (TP + FN)}{TP + TN + FP + FN} \tag{3.28}$$

$$maximum\ index = TP + \frac{1}{2}(FP + FN) \tag{3.29}$$

$$ARI(C, C') = \frac{index - expected\ index}{maximum\ index - expected\ index} \tag{3.30}$$

Figure 3.3b shows that the adjusted Rand index suffers from the same problems as the pF_1 measure, and additionally it drastically downgrades the trivial clusterings where all items are packed into the same cluster (cf. k).

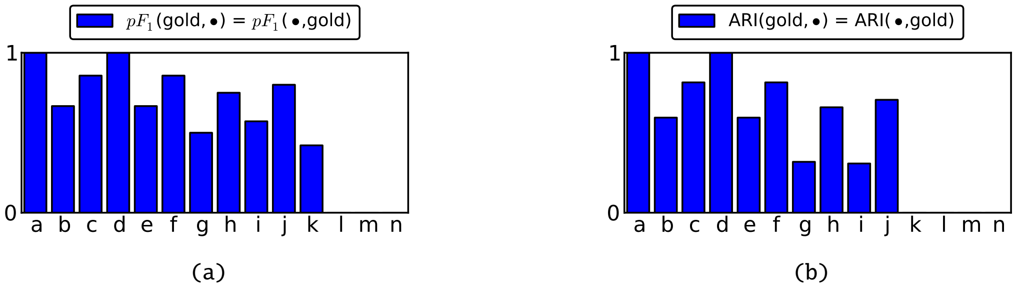


Figure 3.3.: *Pairwise comparison measure results of the synthetic clusterings evaluation. (a) shows the results of the* pairwise F_1 measure*, and (b) shows the results of the* adjusted Rand index.

3.4.4 Information theory based comparison

Information theory based measures — often also called entropy based measures — try to measure the amount of information that is needed for encoding the conversion of one clustering into another. Some basic values on that all information theory based measure rely are listed below.

According to Cover and Thomas (1991), the *entropy* $H(C)$, the *conditional entropy* $H(C|C')$ and the *mutual information* $I(C,C')$ are computed as:

$$H(C) = -\sum_i P(c_i)\log P(c_i)\,, \qquad H(C') = -\sum_j P(c'_j)\log P(c'_j) \tag{3.31}$$

$$I(C,C') = \sum_{i,j} P(c_i,c'_j)\log\frac{P(c_i,c'_j)}{P(c_i)P(c'_j)} \tag{3.32}$$

$$H(C|C') = H(C) - I(C,C')\,, \tag{3.33}$$

where the probabilities $P(c_i)$ and $P(c'_j)$ as well as the joint probability $P(c_i,c'_j)$ are relative frequencies of the contingency table entries:

$$P(c_i) = \frac{n_{i.}}{N}\,, \quad P(c'_j) = \frac{n_{.j}}{N}\,, \quad P(c_i,c'_j) = \frac{n_{ij}}{N}\,. \tag{3.34}$$

LogN Normalized Variation of Information

The *variation of information* (Meilă 2003, 2007) is probably the most prominent information theoretic measure, which is also proven to be a real *metric*[1]. Meilă (2005) extensively analyzes and compares the *variation of information* metric (VI) with various other measures through some important criteria, and arguments that defacto no measure is able to simultaneously satisfy all of some desirable properties, but the variation of information only violates the *boundedness* property, which says that a measure can be projected into $[0,1]$.

Though the *variation of information* does not depend on the size of the dataset or the number of clusters, a normalized version will. Hence Meilă proposes two normalization factors: (a) $\log N$ requiring a fixed size of the dataset, and (b) $2\log K^*$ requiring a maximum number of K^* clusters in C and C' and $K^* \leq \sqrt{N}$. Because we cannot assure the number of clusters to be maximally $\sqrt{N}$, the latter will not be tested here. The VI measure and its $\log N$ normalized variant ($nNVI$) — the n stands for naïve, because now the value depends on the size of the dataset, meaning for instance a value of 0.2 on one dataset may be a good value whereas a value of 0.2 on another dataset may be a rather bad value — is computed as:

$$VI(C,C') = H(C) + H(C') - 2I(C,C') \tag{3.35}$$

$$nNVI(C,C') = \frac{1}{\log N} VI(C,C')\,. \tag{3.36}$$

[1] a metric is a distance function, which satisfies the following conditions: (a) non-negativity, (b) identity, (c) symmetry, (d) triangle inequality (Bronstein et al. 2005).

Since the measure is a distance measure, we need to transform it into a similarity measure by simply using $1 - nNVI$.

The results of the synthetic tests can be reviewed in Figure 3.4a. The orthogonal clustering in m is recognized and downweighted by the $nNVI$ measure, but again the clustering in n where no items are in common with the clustering in a is valued considerably good opposed to the simple changes in g or i.

Normalized Variation of Information

To overcome the "unboundedness problem" of the VI measure in a more sophisticated way, Reichart and Rappoport (2009) proposed the *normalized variation of information* (NVI), "...which guarantees that the score of clusterings that VI considers good lies in [0,1], regardless of dataset size"(p. 1).

$$NVI(C,C') = \begin{cases} \dfrac{VI(C,C')}{H(C)} & \text{if } H(C) \neq 0 \\ H(C') & \text{otherwise} \end{cases} \tag{3.37}$$

Analogously to $nNVI$ the NVI is a distance measure, which means we use $1 - NVI$ in our comparison instead.

In Figure 3.4b the results of the synthetic clustering comparisons are depicted. The NVI-measure is not symmetric, so two values are produced, one for each direction. Note that forcing symmetry can be easily done here. Nevertheless, additionally to the clustering in m, the NVI measure penalizes the clustering in k, and like the $nNVI$ it does not recognize the two different datasets in n.

V-Measure

Another measure from the information theory domain based on entropy is the V measure (V for validity), which was introduced by Rosenberg and Hirschberg (2007), in order to overcome the shortcomings of the VI metric. Remember that the main problem is that VI is not bounded. Rosenberg and Hirschberg formulate the V measure in terms of *homogeneity* and *completeness*; two criteria for distinguishing good clustering results from bad ones, that can be directly computed and weighted.

Homogeneity, on one hand, is satisfied if all clusters of a clustering C contain only those elements that are also grouped into the same cluster in C', and completeness, on the other hand, is satisfied if all clusters of a clustering in C contain all of the elements that are grouped into the same cluster in C'. Since homogeneity and completeness are typically diametral, only the perfect clustering simultaneously satisfies homogeneity and completeness. Homogeneity (h) and completeness (c) are computed as:

$$h = \begin{cases} 1 & \text{if } H(C,C') = 0 \\ 1 - \dfrac{H(C|C')}{H(C)} & \text{otherwise} \end{cases} \quad , \quad c = \begin{cases} 1 & \text{if } H(C',C) = 0 \\ 1 - \dfrac{H(C'|C)}{H(C')} & \text{otherwise} \end{cases} \quad . \tag{3.38}$$

The V measure is then computed as analogously to the F measure as the weighted harmonic mean of h and c, where the default scenario is equal weights:

$$V = F_1 = \frac{2 \times h \times c}{h + c} \, . \tag{3.39}$$

The main advantage of the V-measure is that the criteria for clustering comparison (i.e. homogeneity and completeness) are put into numbers. By weighting them, like the F measure does with precision and recall, it can be configured to best match the task at hand. However, in this thesis completeness and homogeneity are equally important, and besides to that, unequal weighting of homogeneity and completeness yields V to be asymmetric, which is contrary to the symmetry property in Section 3.3.2.

The results of the synthetic test cases are shown in Figure 3.4c. A major problem of the V measure is that it generally favors small clusters (see e.g. Reichart and Rappoport 2009 for a more detailed discussion), which can be seen in l and also as an artifact of cluster conformation (Sec. 3.3.1) in n. Another conspicuous thing is that it delivers the exact same results as the *normalized mutual information*, which is described below.

Normalized Mutual Information

Yet another information theory based measure is the *normalized variation of information* proposed by Strehl (2002). It is a normalized version of the *mutual information* $I(C,C')$ (cf. Eq. (3.32)), which can be seen as the amount of information a clustering C “knows” about another clustering C', where it is the reduction in the uncertainty of one clustering due to the knowledge of the other (Cover and Thomas 1991). According to Manning et al. (2008) the arithmetic mean $[H(C) + H(C')]/2$ is a tight upper bound on $I(C,C')$. Normalizing $I(C,C')$ by this factor yields:

$$NMI(C,C') = \frac{I(C,C')}{[H(C) + H(C')] \, / \, 2} = \frac{2\, I(C,C')}{H(C) + H(C')} \, . \tag{3.40}$$

As stated later by Strehl and Ghosh (2003) another possible $I(C,C')$ normalization is the geometric mean $\sqrt{H(C)H(C')}$ instead of the arithmetic mean. Since the difference between these two factors is not tremendous and the choice is simply one of preference, the arithmetic mean is used throughout this thesis.

Figure 3.4d depicts the results of the synthetic tests, and as figured above the value yields the exact same results as the V measure. Using some transformations, it can be shown that $V = NMI$ in its basic forms. The full derivation of the proof can be found in Appendix A.

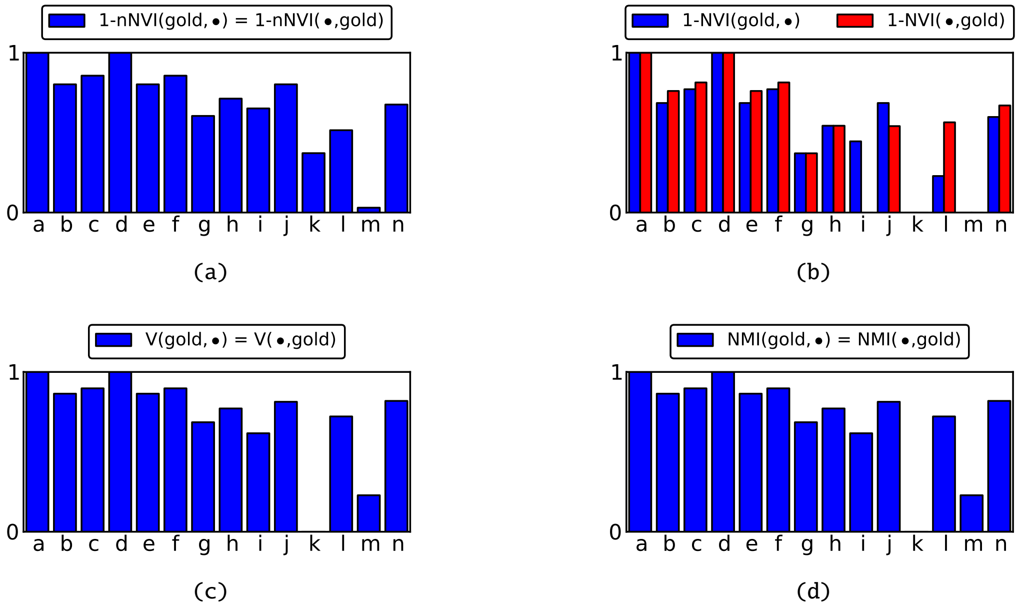


Figure 3.4.: *Information theory based comparison measure results of the synthetic clusterings evaluation. Pairwise comparison measure results of the synthetic clusterings evaluation. (a) shows the results of the* log n normalized variation of information measure*, (b) shows the results of the* normalized variation of information*, (c) shows the results of the* V measure *and, (d) shows the results of the* normalized mutual information.

3.4.5 B^3

Another class of clustering comparison measures (according to Amigó et al. (2009)) is that of B^3 (pron. b-cubed) measures, where for each element in the dataset D precision and recall values are computed and averaged by some weighting scheme. The main advantage of B^3 measures is that precision and recall values of items can be weighted independently, and so, it can be configured to best evaluate the task at hand. However, in this thesis the default weighting scheme is used, i.e. equal weights to every item. A detailed description can be found in (Bagga and Baldwin 1998).

In order to compute the B^3 precision and recall values we need to define a function $cl(\cdot)$ and $cl'(\cdot)$ that takes as input an item $d \in D$ and returns as output the index i or j of the cluster $c_i \in C$, or $c'_j \in C'$ respectively, to which the item d belongs. More formally:

$$\begin{aligned} cl : D \rightarrow \mathbb{N}, d \mapsto cl(d) &:= \{i \in \mathbb{N}_1^K : d \in c_i \wedge c_i \in C \wedge d \in D\} \\ cl' : D \rightarrow \mathbb{N}, d \mapsto cl'(d) &:= \{j \in \mathbb{N}_1^{K'} : d \in c'_j \wedge c'_j \in C' \wedge d \in D\}\ . \end{aligned} \tag{3.41}$$

Using this function, precision, recall and F$_1$ values can be computed utilizing a contingency table as described in Sec. 3.4.1:

$$B^3 p = \frac{1}{|D|} \sum_{d \in D} \frac{n_{cl(d)cl'(d)}}{n_{cl(d).}} \tag{3.42}$$

$$B^3 r = \frac{1}{|D|} \sum_{d \in D} \frac{n_{cl(d)cl'(d)}}{n_{.cl'(d)}} \tag{3.43}$$

$$B^3 F_1 = \frac{2 \times B^3 p \times B^3 r}{B^3 p + B^3 r}\ . \tag{3.44}$$

The results of the synthetic tests in Figure 3.5 show that the B^3 measure behaves similarly to the K measure. It does not recognize missing or additional items, as can be seen in d, and does not penalize clusterings according to property 3. Also, the orthogonal clustering (m) and the two trivial clusterings (k, l) are rated very high in contrast to simple transformations as for example in e and g, which is not the intuitive desired behavior.

It is an interesting observation, that although the computations of K and $B^3 F_1$ are completely different, the scores are nearly the same in our test scenarios. A fact that will not be studied further in this thesis, but should be kept in mind when discussing generally about clustering measures.

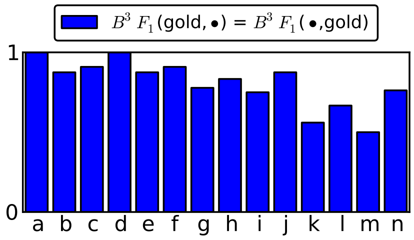


Figure 3.5.: *Results of the evaluation of the synthetic test with the* B^{3}F$_1$ *measure.*

3.4.6 Edit distance comparison

Another class of clustering comparison measures is that based on edit distances. These measures utilize the idea of “editing” a clustering C, until it “reaches” another clustering C', where the individual measures differ in the set of possible edit operations. This methodology is founded on the computation of the well-known *string distance*.

Basic Merge Distance

Menestrina et al. (2010) introduced a linear time algorithm for computing the *generalized merge distance* (GMD), which uses only *splits* and *merges* as possible cluster editing operations. In addition, the GMD is customizable by some *cost function* for splits $f_s(\cdot)$ and merges $f_m(\cdot)$, so that it can be configured to compute various measures e.g. the VI measure or the pF_1 can be computed using the GMD. See Menestrina et al. (2010) for the details. Using a constant factor of 1 for splits and merges (i.e. $f_s(\cdot) = f_m(\cdot) = 1$), gives the *basic merge distance* (BMD), which is both, an intuitive and reasonable score, and thus powerful enough to be considered in this thesis.

Considering $\top$ as the most general clustering of a dataset D, where all elements are grouped into the same cluster, and further considering $\perp$ as the most specific clustering of D, where each element builds its own cluster, the *lattice* between $\top$ and $\perp$ spans all possible clusterings of D. The BMD can be interpreted as the *shortest path* from a clustering C to a clustering C' in the lattice. A basic constraint of the GMD is that merges in C can not create newly clustered elements that are not grouped in the same cluster in C'. This means, split costs are computed before merge costs, which can be thought of as a restriction of possible paths in the lattice. Here, we first need to move down from C (split C) before moving up to C' (merge to C'). For an illustrative example see Figure 3.6, where the lattice is drawn for a dataset $D = \{a, b, c, d\}$ and the example path from a clustering $C = \{\{a, b\}, \{c, d\}\}$ to a clustering $C' = \{\{a, c\}, \{b, d\}\}$ is colored in red.

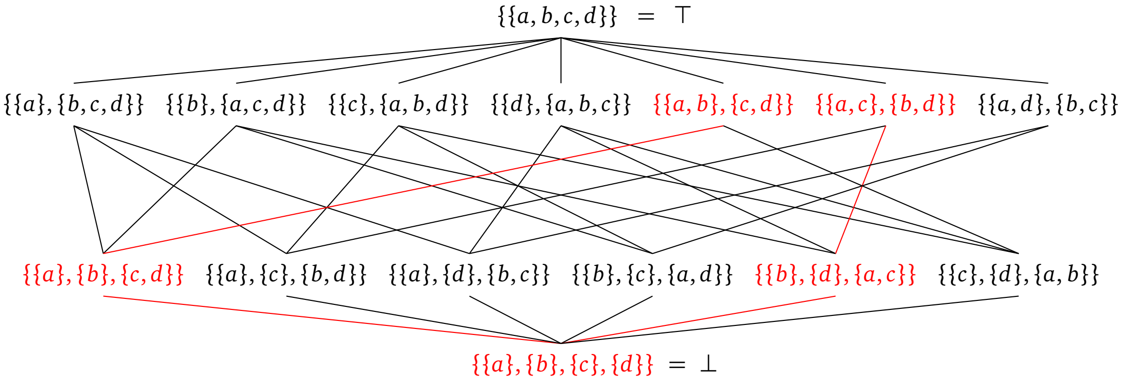

$\{\{a,b,c,d\}\} = \top$
$\{\{a\},\{b,c,d\}\}$
$\{\{b\},\{a,c,d\}\}$
$\{\{c\},\{a,b,d\}\}$
$\{\{d\},\{a,b,c\}\}$
$\{\{a,b\},\{c,d\}\}$
$\{\{a,c\},\{b,d\}\}$
$\{\{a,d\},\{b,c\}\}$
$\{\{a\},\{b\},\{c,d\}\}$
$\{\{a\},\{c\},\{b,d\}\}$
$\{\{a\},\{d\},\{b,c\}\}$
$\{\{b\},\{c\},\{a,d\}\}$
$\{\{b\},\{d\},\{a,c\}\}$
$\{\{c\},\{d\},\{a,b\}\}$
$\{\{a\},\{b\},\{c\},\{d\}\} = \perp$

Figure 3.6.: *The lattice of the most general clustering $\top$ of a dataset to the most specific clustering $\perp$, with a shortest path from the clustering $\{\{a, b\}, \{c, d\}\}$ to the clustering $\{\{a, c\}, \{b, d\}\}$.*

Formalizing the BMD in terms of the GMD according to Menestrina et al. (2010) yields the following definition:

$$BMD(C,C') = GMD_{f_s,f_m}(C,C') \tag{3.45}$$

$$f_s(|X|,|Y|) = 1 \quad , \quad f_m(|X|,|Y|) = 1 \, , \tag{3.46}$$

where X and Y are two connected clusterings on the lattice, and f_s and f_m are the cost functions for splitting X into Y or merging X into Y respectively.

Normalized Basic Merge Distance

The value of the BMD now describes how many operations are needed to get from a clustering C to a clustering C' in a natural and intuitive way, but in order to be conform with property 1 in Section 3.3.2 we need to normalize it to lie in $[0,1]$. The resulting measure is then called *normalized basic merge distance* $(NBMD)$. One option to get compute the $nBMD$ is to divide it by the maximum value the BMD can become for the current underlying dataset:

$$NBMD(C,C') = \frac{BMD(C,C')}{\max(BMD(C,C'))} \tag{3.47}$$

Clearly, the maximum number of operations is the maximum shortest distance of two clusterings in the lattice. Let N be the number of elements in a dataset D, it can be shown that for any clusterings C and C' of the dataset D:

$$\max(BMD(C,C')) = \begin{cases} BMD(\top,\bot) & \text{if } 1 \leq N \leq 3 \\ 1 + BMD(\top,\bot) & \text{if } N > 3 \end{cases} \, , \tag{3.48}$$

which is equivalent to:

$$\max(BMD(C,C')) = \begin{cases} 0 & \text{if } N = 1 \\ N-1 & \text{if } 1 < N \leq 3 \\ N & \text{if } N > 3 \end{cases} \, . \tag{3.49}$$

Since the $NBMD$ is a distance value, $1 - NBMD$ is used instead. By normalizing with the maximum, the $NBMD$ measure is now dependent on the size of the dataset, as is the $nNVI$, which means the value is now hardly interpretable when comparing the results of two datasets.

Figure 3.7a shows the behavior of the measure when computing closeness between the synthetic clusterings described in Figure 3.1. The high values in d and n show that the value suffers from the cluster conformation step in Sec. 3.3.1 again. This is a very big problem of almost every measure evaluated in this thesis.

Fortunately, because of its intuitive interpretability, the BMD can be tuned, so that it penalizes various erroneous differences between C and C', e.g. we could penalize new items in C, i.e. the items in C' that are not present in C before conformation, and vice versa. Possibilities for configurations are endless.

Here, unknown clusters in C and C', i.e. clusters that contain only elements that are unknown to the respective clustering before conformation, will be penalized with a cost factor of 1. As an example consider $C = \{\{a,b\},\{c,d\}\}$ and $C' = \{\{a,e\},\{f\},\{g\}\}$. The number of unknown clusters to C' is 1 since the cluster $\{c,d\}$ consists exclusively of elements that are not in the underlying dataset of C'. Although b is also an element unknown to C' it is embedded in a cluster that contains an element that is known to C' namely a. On the other hand, the number of unknown clusters to C is 2, since the elements f and g build their own clusters, and e is again in a cluster that contains a, which is already known to C. Thus, the $1 - NBMD$ score for the current example is:

$$\begin{aligned} 1 - NBMD(C,C') &= 1 - \frac{BMD(C,C') + \#unknown\ cluster\ C + \#unknown\ cluster\ C'}{max(BMD(C,C'))} \\ &= 1 - \frac{3+2+1}{7} \approx 0.29\,. \end{aligned}$$

Note that without this penalization, the $1 - NBMD$ score is $1 - \frac{3}{7} \approx 0.57$, an unintuitively high value. The normalization by the maximal value of the BMD is not affected by this.

The results of the comparison with the synthetic test clustering are shown in Figure 3.7b. Note that now the missing item of the clustering described in Figure 3.1d is reflected by the score, and also the clustering consisting of only unknown elements (cf. n) is recognized and weighted down as desired. Yet, the scores for the trivial clusterings in k and l are too high in relation to the other values.

3.5 Combining Measures

The results of the analysis of the various clustering comparison measures have shown, that none of the measures mentioned above is fully adequate for the task of comparing lexical chains. Though not a single measure is capable of satisfying all the needs, individual measures satisfy individual needs very well. This leads to the assumption that a combination of measures will best fit our needs.

In this thesis the average of the *ARI* (Sec. 3.4.3) and the $NBMD$ measure (Sec. 3.4.6) will be used for comparing different sets of lexical chains from the same document. For simplicity, the

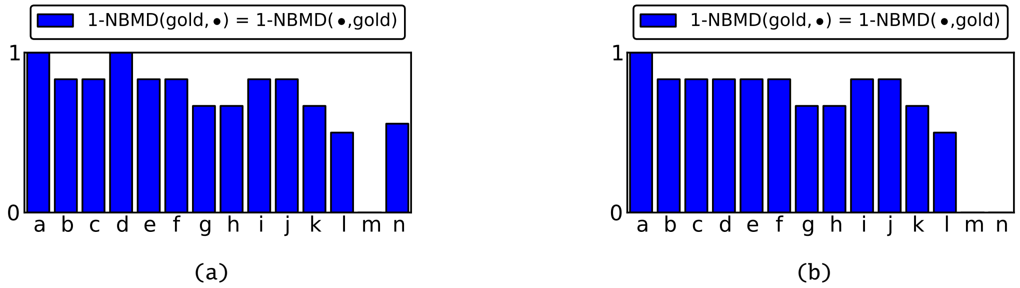


Figure 3.7.: *Results of the synthetic test evaluation using the* normalized basic merge distance. *(a) shows the results of the NBMD , and (b) shows the results of the* NBMD *using the penalizing unknown clusters.*

measure is just called $lccm$ (lexical chain comparison measure), and is defined as the arithmetic mean between ARI and $1 - NBMD$:

$$lccm(C, C') = \frac{1}{2}\left[1 - NBMD(C, C') + ARI(C, C')\right] \ . \tag{3.50}$$

Figure 3.8 shows the scores of the $lccm$. The combination of ARI and $NBMD$ gives no value to the orthogonal clustering in m and to the clustering in n that has no shared items. Additionally it rates down the trivial clusterings, where either all items are grouped into one big cluster (k) or each item builds its own cluster (l). Also, the missing item in d is recognized and not scored as perfect. Still, this measure gives unequal values to simple changes of the same form, which is realized by different heights of the bars in b-j. Ideally these bars will not differ in height, but the analysis from above has shown, that measures satisfying this property come with other problems. So the trick is to find the best trade-off.

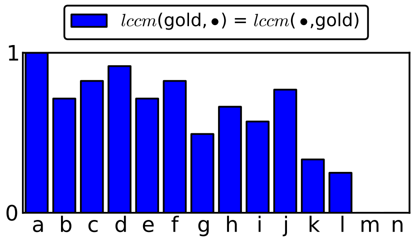


Figure 3.8.: *Results of the evaluation of the synthetic test with the $lccm$.*

Figure 3.9 shows how the individual measures correlate on the synthetic clusterings. It can be seen that the $lccm$ correlates a little with all other measures. This leads to the assumption that we approximated a measure that gets the best of all measures tested above.

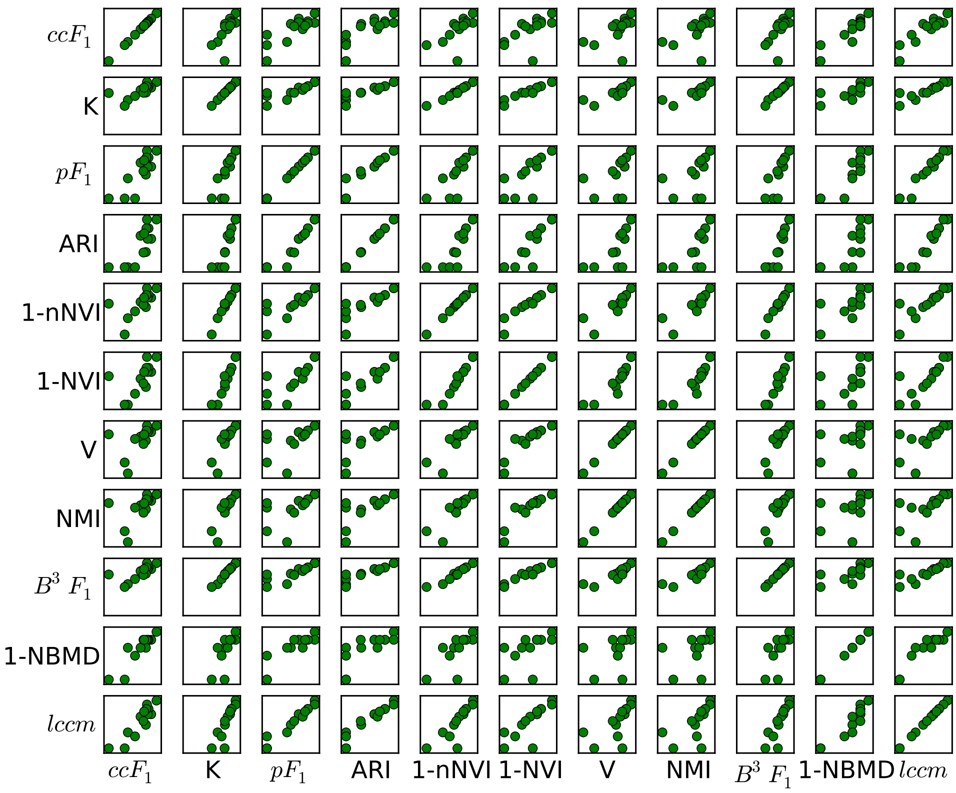


Figure 3.9.: *Correlation of the individual measure with each other based on the results of the synthetic test data. In each plot of a row, the y-axis denotes the measure given in the leftmost plot. In each plot of a column, the x-axis denotes the measure given in the bottommost plot. The correlation between ARI and $1 - NBMD$ for example can be found in row 4 column 10 and vice versa. With perfect correlation all data points lie on the diagonal, as can be seen when comparing the measure with itself.*

4 Annotating Lexical Chains

Text, annotated with lexical chain information is a must-have for many reasons. It can be used either to analyze lexical cohesion in more detail (cf. for example Teich and Fankhauser 2005) in order to build better chaining algorithm, or it can be used for training a supervised lexical chaining algorithm, or, and this is probably the greatest desire, it can be used to qualitatively compare lexical chains of a particular document from many annotating sources, i.e. inter-annotator agreement and evaluation of automatic approaches. Creating such a *gold standard* allows researchers to test their implementation individually on a common basis. This practice is generally applied (if possible) in the *machine learning* area where a high agreement in annotated data is required otherwise the method is not considered to be reliable (Morris 2010).

Unfortunately, the annotation of lexical chains is highly influenced by the subjective interpretation of the text by individual annotators (Morris and Hirst 2004) which also substantiates the fact that currently no gold standard exists. Morris (2010) noted that the difference in interpretation is approximately 40 % and that high inter-annotator agreement is extremely hard to achieve and — due to natural human text interpretation — is either not possible or just the result of unnaturally forced interpretation scenarios.

Nevertheless, Hollingsworth and Teufel (2005) as well as Cramer et al. (2008) experimented with the annotation of lexical chains; both concluding with the same result that high inter-annotator agreement cannot be achieved.

In the context of this thesis an annotation project was carried out in which a corpus was annotated with lexical chain information. One hundred general domain news articles from a German newspaper were annotated by two annotators. The next sections describe the underlying corpus and the applied methodology as well as the toolkit used.

4.1 Corpus

In this thesis the documents of the SALSA[1] 2.0 (Burchardt et al. 2006) corpus were chosen to form the basis for the annotation of lexical chain information. SALSA 2.0 itself is based on the semi-automatically annotated TIGER Treebank 2.1 (Brants et al. 2002). Salsa already provides in-

[1] SALSA II - The Saarbrücken Lexical Semantics Acquisition Project

formation, such as lemmas, part-of-speech tags, syntactic structure or FrameNet[2] frame structure. The documents are general domain news articles from a German newspaper comprising about 1,550 documents and around 50,000 sentences in sum though not all documents are appropriate for lexical chaining. The coverage of SALSA sentences to Tiger sentences is ∼48 %, and the average coverage of sentences per document is ∼46%. It is thus crucial to not only rely on the SALSA data, but on the Tiger data as well since only complete texts can be considered for annotation, as for cohesion the completeness the text is of utmost importance. Figure 4.1 shows the relative coverage of sentences of a SALSA documents in its Tiger counterpart.

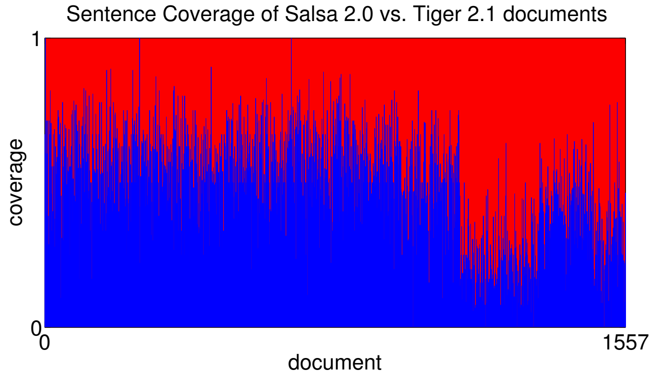


Figure 4.1.: *Coverage of Salsa and Tiger sentences per document. The blue area shows the coverage of each Salsa document to its corresponding Tiger document, and the red are is thus the relative amount of missing sentences in Salsa. Note that only three SALSA documents cover 100 % of the sentences of the respective Tiger document.*

The long time goal is to enhance all documents with lexical chain information but for the time being 100 documents are annotated that lie around the median of 275 tokens. Figure 4.2 shows the length of each document on a logarithmic scale. Some of the documents are considered inappropriate because of wrong document boundaries or inappropriate type. Inappropriate types of documents are those that contain more than one text, e.g. news ticker like texts. These documents were heuristically identified and discarded since the automatic approaches rely on correct document boundaries for training as well as for testing. Appropriate documents can thus be seen as documents that contain exactly one coherent and cohesive text.

[2] http://framenet.icsi.berkeley.edu – The Berkeley FrameNet Project (Baker et al. 1998)

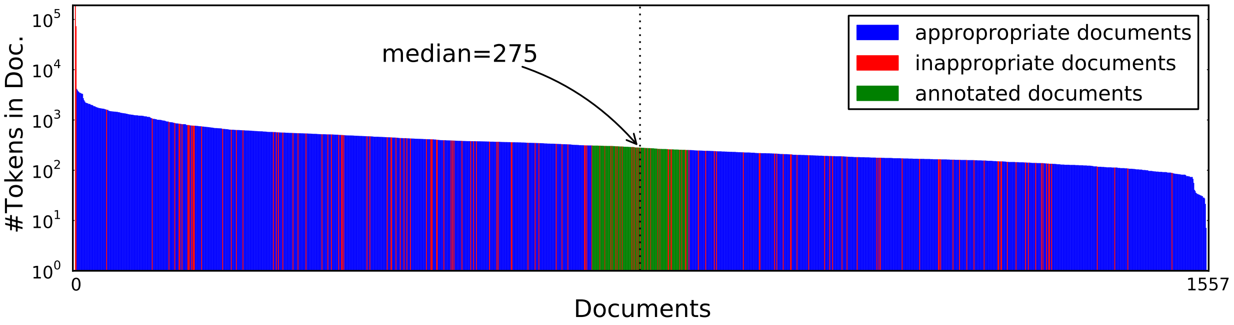


Figure 4.2.: *Tiger 2.1 documents sorted by the number of their tokens on a logarithmic scale. Red bars describe documents that are considered inappropriate, green bars describe documents that have been annotated during the annotation project, and blue bars describe the remaining documents.*

4.2 Annotation Scheme

In order to minimize the subjective choices of the different annotators, *annotation guidelines* were developed comprising a total of ten pages; including a short introduction into lexical chains, the actual annotation scheme and some examples. These guidelines can be found in Appendix B.

Additionally the candidates for lexical chaining, i.e. the possible constituents of the lexical chains are preselected by a simple heuristic based on a part-of-speech pattern. This pattern preselects candidate words or expressions by considering only nouns, noun-noun combinations and adjective-noun combinations.

Opposing to the original definition of lexical chains — recall that any kind of word or phrase is defined to be a possible constituent of lexical chains — some simplifications were applied that made the annotation process more tractable. In the context of this thesis only nouns, noun compounds and non-compositional adjective noun phrases like “dirty money” are considered as candidates for lexical chaining. This strategy strongly coheres with the procedure of Cramer et al. (2008) and Hollingsworth and Teufel (2005).

Because there is no consensus among researchers about what actually constitutes a lexical semantic relation (Cramer et al. 2008), the term *dense chain* is introduced which describes basically a special type of lexical chain in which every element is related to every other element that is also contained in the same chain. Transitivity is thus eliminated. Terms are considered to be related if they share the same topic, i.e. common sense and knowledge of the language is needed to decide which terms belong together in the same topic and whether a chosen topic is neither too broad nor too narrow. A single dense chain can thus be assigned a definite topical description of its items.

Yet, the choice of the best chain for a lexical item is often fuzzy. Hollingsworth and Teufel approached this problem by allowing a lexical item to occur in different lexical chains. This thesis on the other hand follows the concept of *cohesive harmony* introduced by Hasan (1984) where complete chains can be linked in order to achieve the cohesive harmony. For this purpose, the concept of a so-called *level two links*, which is nothing else but a cohesive tie between two lexical items that exist in distinct dense chains, is introduced. Having such a link between two chains, both chains can be assigned a topical description which is broader than the description of the individual chains. Mind that by doing so multiple chains each describing a narrow concept can be linked to form a single broader concept. More details can be found in the annotation guidelines in Appendix B.

4.3 Annotation Toolkit

Morris and Hirst (2004) annotated and analyzed lexical chains with paper and pen. This is obviously not an option here; the maintenance of multiple annotators and multiple texts is too expensive. Hence a tool is sought where a lot of annotations are easy to maintain and the tool must be easy to handle even for non-experts.

Stührenberg et al. (2007) introduced SERENGETI[3], a tool for annotating i.a. syntax, anaphora, and lexical chains. Serengeti is a web application allowing a server sided management of documents, annotations, and user groups in a central server sided place. Unfortunately it is not publicly available.

Another tool that is also designed as web application but is freely available is BRAT[4] (Stenetorp et al. 2012). Brat also allows for server sided management of documents and annotations as well as user management, but allows also for local annotation projects. Additionally, it is customizable in whatever you want to annotate by providing some basic elements like *entities, relations, attributes* and *events*. Lexical chain information can not be annotated from scratch but can easily be configured. Unfortunately the tool did not become available early enough to be used in this work.

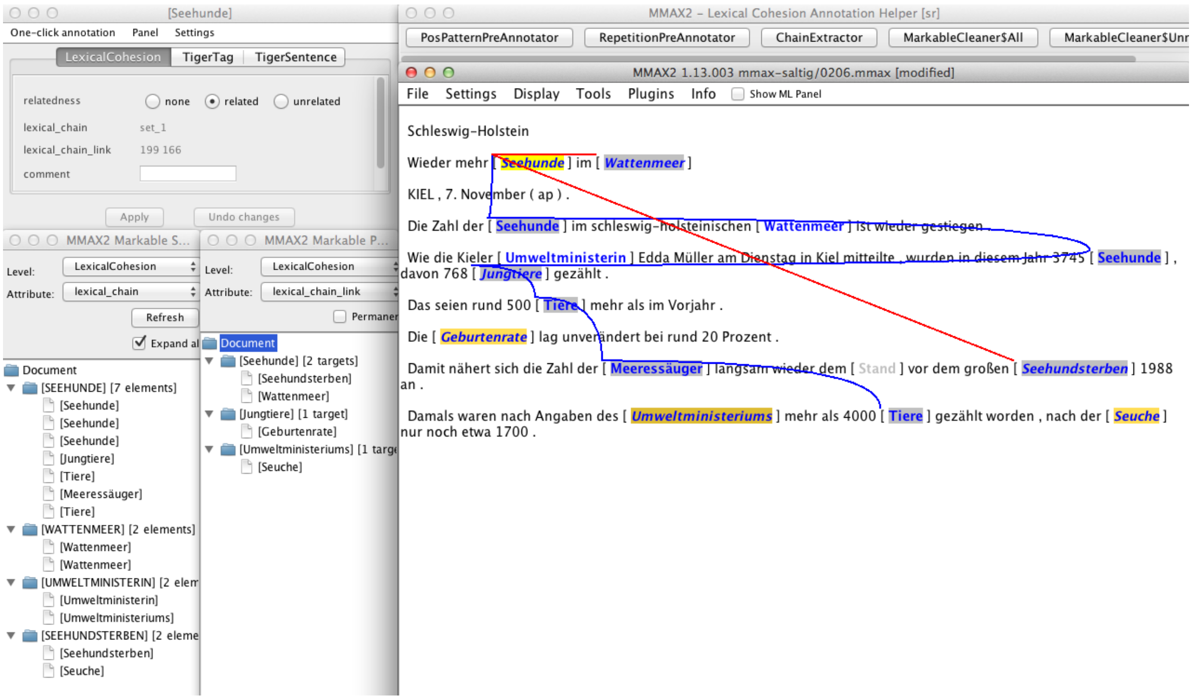
Another widely accepted tool is MMAX2[5] (Müller and Strube 2006) which is also open source and freely available. MMAX2 is fully customizable via xml and xsl files, i.e. the kind of annotation as well as the look-and-feel is fully customizable which makes it flexible and sort of complex to configure at once. Unfortunately MMAX2 is a local application which means that the documents

[3] http://coli.lili.uni-bielefeld.de/serengeti/

[4] http://brat.nlplab.org

[5] http://mmax2.sourceforge.net or http://mmax2.net

as well as the annotations are stored locally on each annotator's machine. Though, the handling of multiple annotators is possible with a little effort. MMAX2 is equipped with a Java API and thus allows for the automatic processing of MMAX2 projects[6]. Additionally, it allows for the extension of self-made plugins. These reasons lead to the decision for using MMAX2 as the toolkit for the annotation project accompanying this thesis. Figure 4.3 shows a screenshot of the basic usage of MMAX2 while annotating a document.



relatedness ○ none ● related ○ unrelated
lexical_chain set_1
lexical_chain_link 199 166
comment

Level: LexicalCohesion
Attribute: lexical_chain

Level: LexicalCohesion
Attribute: lexical_chain_link

Schleswig-Holstein

Wieder mehr [Seehunde] im [Wattenmeer]

KIEL , 7. November (ap) .

Die Zahl der [Seehunde] im schleswig-holsteinischen [Wattenmeer] ist wieder gestiegen .

Wie die Kieler [Umweltministerin] Edda Müller am Dienstag in Kiel mitteilte , wurden in diesem Jahr 3745 [Seehunde] , davon 768 [Jungtiere] gezählt .

Das seien rund 500 [Tiere] mehr als im Vorjahr .

Die [Geburtenrate] lag unverändert bei rund 20 Prozent .

Damit nähert sich die Zahl der [Meeressäuger] langsam wieder dem [Stand] vor dem großen [Seehundsterben] 1988 an .

Damals waren nach Angaben des [Umweltministeriums] mehr als 4000 [Tiere] gezählt worden , nach der [Seuche] nur noch etwa 1700 .

Document
- [SEEHUNDE] [7 elements]
  - [Seehunde]
  - [Seehunde]
  - [Seehunde]
  - [Jungtiere]
  - [Tiere]
  - [Meeressäuger]
  - [Tiere]
- [WATTENMEER] [2 elements]
  - [Wattenmeer]
  - [Wattenmeer]
- [UMWELTMINISTERIN] [2 elem
  - [Umweltministerin]
  - [Umweltministeriums]
- [SEEHUNDSTERBEN] [2 eleme
  - [Seehundsterben]
  - [Seuche]

Document
- [Seehunde] [2 targets]
  - [Seehundsterben]
  - [Wattenmeer]
- [Jungtiere] [1 target]
  - [Geburtenrate]
- [Umweltministeriums] [1 targe
  - [Seuche]

Figure 4.3.: *Screenshot of the* MMAX2 *user interface. In the center region of the screenshot the word "Seehund" is selected; the blue lines describe the current chain members that are in the same chain as the selected word; the red lines describe level 2 links to other lexical items that are in distinct chains; the buttons on the top belong to a plugin that provides some helper functions; the upper left frame show some properties of the currently selected word; and both of the lower left frames visualize a summary of chained and linked lexical items.*

Details about the usage with respect to the current annotation scheme and corpus can be found in Appendix C the *Lexical Chain Annotation Guidelines Companion*.

[6] In MMAX2 a document is stored as an MMAX2 project. A corpus or a collection of projects may share the same definition and layout files.

4.4 Lexical Chain Analysis

One hundred general domain news articles from a German newspaper were manually annotated independently by two annotators in the context of this thesis which will be analyzed in this section. Table 4.1 shows some interesting facts about the number of identified chains and links, as well as the average size of the chains.

Table 4.1.: *Manually annotated lexical chains in numbers. Merged chains represent those chains that are merged because of a level two link between one of its members each.*

	Annotator A	Annotator B
average number of lexical items per document	38.66	38.96
average number of chains per document	11.25	7.38
average number of links per document	5.47	2.41
average size lexical chains	3.69	5.57
average number of merged lexical chains	6.10	4.99
average size merged lexical chains	7.60	8.91

First of all, the average number of lexical items per document is nearly the same. This raises the assumption that both annotators perceived the same lexical items as related for the current text. This can be traced back to the fact that candidate items were preselected using a part-of-speech pattern and the annotators only had to decide whether a specific candidate is related to the text or not. On the other hand, a value of 81 % in the average agreement of lexical items (cf. Figure 4.4) shows that even though the choice of lexical items is limited to nouns and adjective noun phrases only, the attitude of the annotators regarding the choice of related and unrelated items is somewhat different. This observation coheres with Morris and Hirst's (2004) findings of 63 % average pairwise agreement.

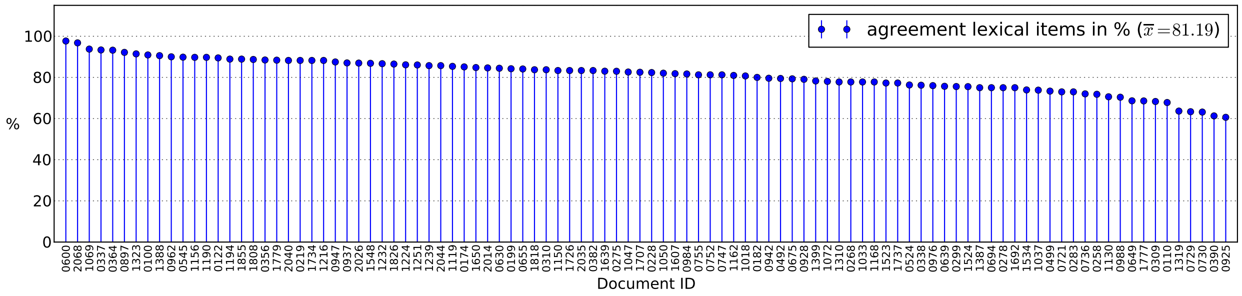


Figure 4.4.: *Agreement of lexical items annotated by annotator A and annotator B as a percentage of lexical items annotated by annotator A or annotator B. The average agreement is 81 %.*

Interesting is also that annotator A creates in average smaller but more chains and also more links between chains than annotator B. Deeper interpretation of this needs further investigation by comparing the sets of lexical chains of each text individually. Figure 4.5 shows the annotator agreement on the individual documents using the before developed *lccm* (cf. Sec. 3.5). The documents are sorted by the agreement of their identified lexical items, i.e. the same sorting as in Figure 4.4. In order to somehow use the level two link information the figure also shows a second agreement score which applies on the sets of lexical chains of each annotator where the original chains are merged via level two links. E.g. when having two chains {“tires”, “driver”, “gasoline”, “race car”} and {“winner”, “loser”, “cup”, “prize”} and a level two link between “race car” and “cup” the merged chain which is then used for evaluation is {“tires”, “driver”, “gasoline”, “race car”, “winner”, “loser”, “cup”, “prize”}.

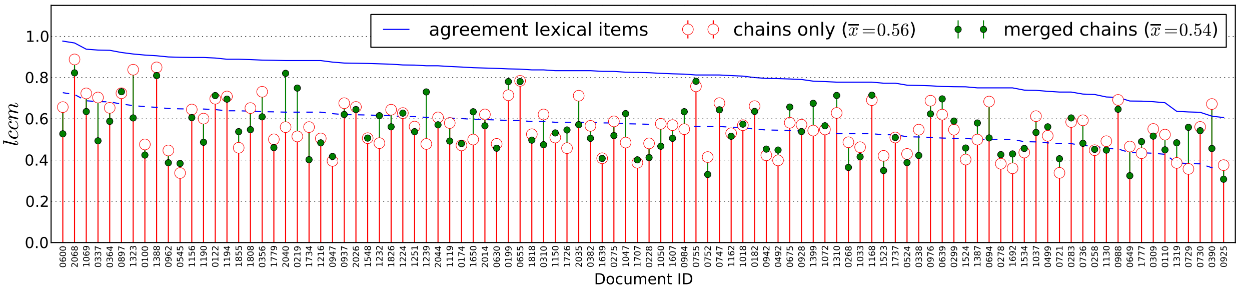


Figure 4.5.: *Individual annotator agreement scores on the complete test set of 100 documents sorted by their agreement in the used lexical items. The red circles show the agreement of the individual chain sets and the green dot shows the agreement of the annotators based on the sets of merged chains. The dashed blue line is just the continuous blue line shifted down by a certain amount, which serves just for illustrative purposes.*

The agreement scores of the assignment of lexical items to lexical chains depend partially on the agreement scores of the identified lexical items themselves. This is a desired behavior since a good lexical chain comparison measure should reflect that. A perfect agreement was never achieved on the data. This shows that the interpretation of a document by different annotators is somewhat different even with concise guidelines. But this also shows that an ultimate gold standard for a corpus with lexical chain information can not be easily compiled.

Although one could initially expect a better agreement when merging chains — consider that by merging two chains two ideas are merged to a single broader idea and chances are that broader ideas are captured equally by different annotators — in average the agreement does not increase, rather it slightly decreases.

5 Statistical Methods for Lexical Chaining

With growing computational power and even more available data more and more methods emerge or revive that exploit statistical assumptions about natural language. According to the “law of large numbers” statistics shows its full strength once enough data is available and moreover processable. In earlier works the input data is often restrained due to technical limitations in time and space, but the faster the computations become and the more data can be accessed efficiently, the more data can be processed in the same period and the more accurate and authentic the results become.

Most statistical tests in natural language processing are based on the occurrence and co-occurrence of terms in the same document, paragraph, sentence, n-gram, or the like. This thesis employs two well-studied statistical methods for creating something that Barzilay (1997) called an *automatic thesaurus* which will then be adapted for lexical chaining.

5.1 Candidate Lexical Items

A candidate lexical item is a word or phrase that will be eventually inserted into a lexical chain or not. In the following, candidate lexical items are preselected in a text by the same heuristic that is also applied in Chapter 4 for the simplification of the annotation process. The heuristic is a regular expression working on patterns of the part-of-speech of the words which preselects all nouns, noun-noun combinations, adjective-noun combinations as well as adjective-noun-noun combinations. This way, the heuristic tries to identify meaningful noun compounds and non-compositional adjective noun phrases. As an example consider the sentence in 5.1.

(5.1) The child called for an exciting bedtime story.

Here, the following words and phrases are considered as possible candidate lexical items: {“child”, “story”, “bedtime story”, “exciting bedtime”, “exciting bedtime story”}.

5.2 Probabilistic Topic Models

Topic models (TMs) are a suite of unsupervised algorithms designed for unveiling some hidden structure in large data collections. The basic assumption of TMs is that there exists at least some latent theme in the data that actually generated it. In natural language processing, the data is often represented as *document-term matrix* where rows are documents, columns are vocabulary terms, and the value in a specific row and column represents the number of occurrences of the term in the document. The key idea is that documents can be represented as composites of so-called *topics* where a topic itself represents as a composite of words. Still, TMs are not bound to documents and terms — although the name might suggest that — they are more generally applicable to any kind of discrete data. The word "topic" simply emerged because these algorithms, when applied on language data, deliver results that remind one of groups of related terms that share the same topic (Blei et al. 2003).

Topic models have its origin in the psychology community where the assumption is that the application to large datasets can yield insights into human cognition (Steyvers and Griffiths 2006). It all began with *latent semantic analysis* (LSA) (Deerwester et al. 1990) — in the information retrieval community also known as *latent semantic indexing* (LSI) — which is not yet a topic model though highly related to the development of these. The LSA method is based on simple linear algebra: Documents and words are represented as points in the Euclidean space. By applying matrix factorization a lower dimensional view of the training data is created that might reveal some shared structure such as synonymy or polysemy. Unseen documents can then also be transformed into a lower dimensional view by multiplying with these factors.

When Hofmann (1999a, 1999b) published *probabilistic latent semantic analysis* (pLSA) — also known as *probabilistic latent semantic indexing* (pLSI) or *the aspect model* — he introduced a solid statistical foundation and a proper definition of a generative data model which reveals similar information like LSA but with the values being intuitively interpretable. This makes pLSA the first true topic model. Here, a topic is defined to be a probability distribution over words and a document is defined to be a probability distribution over a fixed set of topics. A word w in a document d is then generated by a statistical mixture model which can be represented as:

$$P(w,d) = P(d)P(w|d) \text{ , with} \tag{5.1}$$

$$P(w|d) = \sum_{i=1}^{T} P(w|z_i)P(z_i|d) \, , \tag{5.2}$$

where T is the number of topics, $P(w|z_i)$ is the probability of the word w being generated by topic z_i and $P(z_i|d)$ is the probability of topic z_i for the current document d. Note however, that z is a latent variable which is unknown. Further, by generating the observation pairs (w,d) independently the bag-of-words assumption is applied which says that the order of words in a document is unimportant. Hofmann also provides a MAP estimate for $P(w|z)$, $P(d|z)$ and $P(z)$ — consider that by applying Baye's rule $P(w,d)$ can be rewritten as $\sum_{i=1}^{T} P(z_i)P(w|z_i)P(d|z_i)$.

However, as noted by Blei et al. (2003) this estimate does not capture unseen documents; it learns the topic mixtures only for the documents it is trained on. Hence, they introduced another TM which is called *latent Dirichlet allocation* (LDA)[1] — according to Blei and Lafferty (2009a) the simplest TM. This topic model now allows for the application of unseen documents by putting a Dirichlet prior on the per-document topic distribution. The Dirichlet distribution is a valid choice since it is a conjugate prior of the multinomial distribution which is an important fact for the mixture model assumption. Girolami and Kabán (2003) noted that the LDA model is equivalent to the pLSA model when using a uniform Dirichlet prior. Though, by placing a non-uniform Dirichlet prior on the topic distribution, LDA overcomes the global topic proportion of pLSA and conditions the topic probabilities on the individual document a word belongs to.

In the following, the terminology listed in Table 5.1 is used which is equivalent to the terminology used in Griffiths and Steyvers (2004) and Steyvers and Griffiths (2006). Figure 5.1 shows the graphical model representation of LDA and Listing 5.1 depicts the corresponding generative process, i.e. how a document is created.

Table 5.1.: *LDA Terminology.*

shorthand notation	description
T	number of topics
D	number of documents
V	size of the vocabulary
N_d	number of words in document d
$P(w\|z)$, $\phi^{(z)}$	probability distribution over words for a particular topic z
$P(z\|d)$, $\theta^{(d)}$	probability distribution over topics in a particular document d
β	Dirichlet hyperparameter for estimating ϕ
α	Dirichlet hyperparameter for estimating θ

The Dirichlet parameter α can be seen as a controlling mechanism for the a priori probability of the per-document topic distribution with a lower α being the distribution more sparse but when too low there is danger of overfitting. β on the other hand is the Dirichlet parameter for the a priori probability of the per-topic word distribution, again with a lower β being the distribution more specific. The variables of interest are θ and ϕ — note that $P(w|d)$ is equivalent to $P(w|\theta, \phi)$ when θ and ϕ are known. For obtaining these variables one would need to reverse the generative process in Listing 5.1. As discussed in Blei et al. (2003) the computation of this is intractable. Thus, they use *mean field variational inference* for estimation, though other estimation methods are also being used e.g. *collapsed variational inference* (Teh et al. 2007) or

[1] The name “latent Dirichlet allocation” is built of the words “latent” because the topic structure is hidden in the document collection and must be estimated, “Dirichlet” because the used probability distribution is a Dirichlet distribution and “allocation” because in every iteration the result of the Dirichlet is used to allocate the words of the document to different topics (Blei 2012).

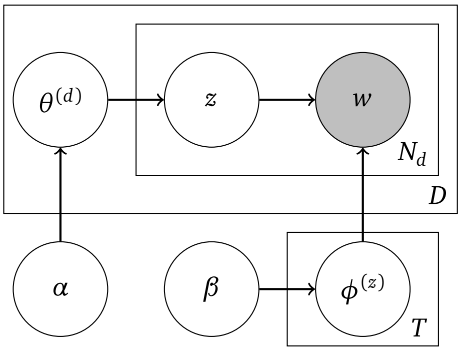


Figure 5.1.: *A graphical model of LDA.*

Listing 5.1: *The generative process of LDA.*

```
/* topic plate */
for each $z \in [1,T]$ do
    draw per-topic word distribution : $\phi^{(z)} \sim \mathrm{Dir}(\beta)$
end for
/* document plate */
for each $d \in [1,D]$ do
    draw per-document topic proportion: $\theta^{(d)} \sim \mathrm{Dir}(\alpha)$
    /* word plate */
    for each $i \in [1,N_d]$ do
        sample a topic assignment: $z_i \sim \mathrm{Mult}(\theta^{(d)})$
        sample a word from topic: $w_i \sim \mathrm{Mult}(\phi^{(z_i)})$
    end for
end for
```

stochastic variational inference (M. Hoffman et al. 2012). This thesis focuses on the *Gibbs sampling* method as proposed by Griffiths and Steyvers (2004).

The *Gibbs sampling* methodology for LDA as described by Griffiths and Steyvers does not explicitly model the parameters ϕ and θ as the parameters to be estimated, rather the posterior distribution of words to topics $P(z|w,d)$ is considered from which the estimates θ and ϕ can then be obtained. Gibbs sampling is an algorithm from the suite of *Markov chain Monte Carlo* methods (MCMC) which samples from a multivariate distribution by considering only one dimension at a time and conditions on the other dimensions from the last stable Markov state. Hence, a Markov chain is created. Generating a large enough collection of samples then approximates the posterior. Sampling from $P(z|w,d)$ can be approximated by:

$$\mathbf{z}_{(d,w)} \sim P(\,\mathbf{z}_{(d,w)} = j \,|\, \mathbf{z}_{-(d,w)}, w, d) \propto \frac{C^{VT}_{wj} + \beta}{\sum\limits_{n=1}^{V} C^{VT}_{nj} + V\beta} \times \frac{C^{DT}_{dj} + \alpha}{\sum\limits_{m=1}^{D} C^{DT}_{mj} + T\alpha}\,, \tag{5.3}$$

where $\mathbf{z}$ is a vector of topic assignments referring to all word tokens in all documents, w is a particular word and d a particular document, $\mathbf{z}_{(d,w)}$ is the current topic assignment for w in d and $\mathbf{z}_{-(d,w)}$ are all topic assignments without the assignment of (d,w), C^{VT} is a $V \times T$ count matrix with C^{VT}_{wj} being the number of times a word w is assigned to topic j excluding the current instance (d,w), C^{DT} is a $D \times T$ count matrix with C^{DT}_{dj} being the number of times any word in document d is assigned to topic j excluding the current instance (d,w) and α and β are the respective Dirichlet hyperparameters. Note however, that Equation (5.3) is just an approximation which gives the unnormalized probability. For further details and a full derivation of Equation (5.3) as well as the Gibbs sampling method for LDA in general see Griffiths and Steyvers (2004); Steyvers and Griffiths (2006); Heinrich (2009). After a valuable burn-in phase, meaning letting the algorithm run some iterations without collecting samples, samples can be collected and the probability distributions ϕ and θ can be approximated as:

$$\phi^{(z)}_{w} \propto \frac{C^{VT}_{wz} + \beta}{\sum\limits_{n=1}^{V} C^{VT}_{nz} + V\beta}\,, \qquad \theta^{(d)}_{z} \propto \frac{C^{DT}_{dz} + \alpha}{\sum\limits_{m=1}^{D} C^{DT}_{mz} + T\alpha}\,. \tag{5.4}$$

The LDA method is widely accepted in the machine learning domain and has been proven to deliver good results with an acceptable computational complexity. Also, a variety of improvements to the ordinary LDA exist: e.g. to overcome the bag-of-words assumption (Wallach 2006; Wang et al. 2007; Blei and Lafferty 2007, 2009b), to automatically estimate the number of topics (Teh et al. 2006), to organize topics hierarchically (Blei et al. 2004), to track topics over time (Wang and McCallum 2006) or to speed up the computation by parallelizing it (Newman et al. 2009). Additionally a number of freely available or open source implementations exist for the vast LDA variety. In the context of this thesis the GibbsLDA++[2] framework (Phan and Nguyen 2007) as well as the JGibbLDA[3] framework from the same authors were used and adapted to the current needs. Based on the information that the standard Gibbs LDA unveils, three algorithms for lexical chaining have been developed which are explained in detail in the following sections.

[2] http://gibbslda.sourceforge.net

[3] http://jgibblda.sourceforge.net

5.2.1 LDA Mode Method (LDA-MM)

The LDA-MM approach places all word tokens that share the same topic id into the same chain. The point is now how to decide to which topic a word belongs to. The most straightforward approach is to collect a single sample $z \sim P(z|w,d)$ and use this assignment for chaining. According to Riedl and Biemann (2012), a single sample is accompanied by a strong variance. Thus, they propose two other techniques for getting a good topic assignment with less variance. The main idea of both methods is to collect a number of samples from $P(z|w,d)$ and then pick that topic id z that was assigned the most. A first methodology collects each last sample of multiple Markov chains with the same initialization and a second methodology collects multiple samples during the run of a single Markov chain. The latter is called the *mode method* and is also applied here.

More formally, let $\mathbf{s}^{(d,w)}$ be the vector of assignments that have been collected for a certain word w in a certain document d with each $s_i^{(d,w)}$ referring to the i-th sampled topic id for (d,w). In other words, regarding Equation (5.3) let $s_i^{(d,w)}$ be the $\mathbf{z}_{(d,w)}$ in the i-th sampling step. Further let $z^{(d,w)}$ be the topic id that was most assigned to the word w with respect to the samples in $\mathbf{s}^{(d,w)}$. Precisely, $z^{(d,w)}$ is defined to be the mode in $\mathbf{s}^{(d,w)}$ — in case of multiple modes a random mode is chosen, though this never happened in practice during this thesis either.

$$\begin{aligned} z^{(d,w)} &= \text{mode}\,(\mathbf{s}^{(d,w)}) && (5.5)\\ &\approx \arg\max_j\,(P(z=j|w,d)) && (5.6) \end{aligned}$$

The LDA-MM assigns for every word w which is a candidate lexical item of a certain document d which is assigned the same topic $z^{(d,w)}$ to the same chain; hence implicitly disambiguating the terms.

Level Two Link Extension

The possibility to create level two links is given by taking the second most occurring topic for a given word if it exceeds a certain threshold. Consider an example: A certain candidate item with topic 4 as its most occurring topic is chained together with all other lexical items also with topic 4 as their most occurring topic. If topic 2 occurred the second most in the sample set for that word and the relative number of occurrences for that topic exceeds for example a fraction of 0.3 then the word is linked to any word where topic 2 occurred most. Here it suffices that only one link is created instead of tying the candidate item with all other words with topic 2 because topics / lexical chains represent ideas and ideas need to be tied only once. According to this, a second link from items with topic 4 to items with topic 2 can be omitted.

5.2.2 LDA Graph Method (LDA-GM)

The LDA-GM algorithm approaches the chaining problem slightly different: First it creates a similarity graph based on the comparison of topic distributions for given words and then applies a clustering algorithm in order to find semantically related words.

Let $\psi^{(d,w)}$ be the per-word topic distribution $P(z|w,d)$. Analogously to the LDA-MM, $\psi^{(d,w)}$ can be obtained by counting how often a certain topic id z occurs in the sample collection $\mathbf{s}^{(d,w)}$ for a particular word w and document d.

The semantic relatedness between two words w_i and w_j can then be measured by their similarity score of the topic distributions $\psi^{(d,w_i)}$ and $\psi^{(d,w_j)}$. For every candidate lexical item in a document the similarity with every other word in the document is computed which results in a matrix containing a similarity value for every candidate pair in the document as depicted in Table 5.2.

Table 5.2.: *Similarity matrix, with sim_{ij} being a similarity value in $[0,1]$ and $sim_{ij} = 1$ if $\psi^{(d,w_i)}$ and $\psi^{(d,w_j)}$ are considered similar.*

$$\begin{array}{c c} & \begin{array}{cccc} w_1 & w_2 & \cdots & w_{N_d} \end{array} \\ \begin{array}{c} w_1 \\ w_2 \\ \vdots \\ w_{N_d} \end{array} & \begin{pmatrix} 1 & sim_{12} & \cdots & sim_{1N_d} \\ sim_{21} & 1 & \cdots & sim_{2N_d} \\ \vdots & \vdots & & \vdots \\ sim_{N_d 1} & sim_{N_d 2} & \cdots & 1 \end{pmatrix} \end{array}$$

Using graph theory, this matrix can also be interpreted as an adjacency matrix with candidate items being nodes and edges being weighted with the similarity value sim_{ij} for any two nodes $i,j : i \neq j \land i,j \in \{1,2,\ldots,N_d\}$. Note that if the similarity measure is symmetric i.e. $sim_{ij} = sim_{ji}$ then only the upper or lower triangular matrix needs to be computed and the graph becomes undirected.

Two different similarity measures will be tested, both of which are symmetric:

1. Euclidean dissimilarity: $sim_{ij} = 1 - ||\psi^{(w_i)} - \psi^{(w_j)}||$. Note that normalization is not needed here since $\psi^{(w_i)}$ and $\psi^{(w_j)}$ are probability distributions and thus $\sum_k \psi_k^{(w_i)} = \sum_k \psi_k^{(w_j)} = 1$.
2. cosine similarity: $sim_{ij} = \psi^{(i)} \cdot \psi^{(j)}$.

Let $G = (V,E)$ be the graph representation of a document with vertices $V = \{v_1,\ldots,v_{Nd}\}$ and weighted edges $E = \{(v_1,v_2,sim_{12}),\ldots(v_{Nd},v_{Nd-1},sim_{N_d N_d -1})\}$, where sim_{ij} is either the cosine or Euclidean similarity. Because of a symmetric similarity matrix, the graph is being treated as undirected. Graph clustering can be now applied in order to find the groups of words that show a strong level of similarity in their topic distributions. The basic assumption is that each cluster groups together those words that are semantically related. The *Chinese Whispers*[4] (CW) graph clustering algorithm (Biemann 2006) as a special case of Markov Clustering (MCL) algorithms is applied to find the clusters. A sketch of the algorithm is shown in Listing 5.2 where c_i is defined to be the class label of a vertex v_i.

Listing 5.2: *The Chinese Whispers algorithm. Vertices with the same class label build the final clusters.*

```
/* initialize class labels */
for $i = 1$ to $|V|$ do
    $c_i \leftarrow i$
end for
/* convergence is reached if either
 *  - the algorithm ran a predefined maximum number of iterations, or
 *  - the number of changes is smaller than some predefined threshold or zero */
until $convergence$ do
    /* update class labels */
    for each $v_i \in V$, randomized−order do
        $c_i \leftarrow$ predominant_class_in_neighborhood($v_i$)
    end for
end for
```

The key idea of MCL algorithms is the modeling of a random walker that will most likely not leave a certain group of vertices, i.e. clusters. The property which is thus exploited is called the Small World[5] property. The assumption is now that in our scenario a Small World of semantically related terms is created. Small World Graphs are sparsely connected with a lot of strongly connected components. The current graph is densely connected, i.e. each vertex is connected with each other vertex. Hence, in order to create a Small World graph and thus to guarantee good functionality of Chinese Whispers, the weights of the graph's edges are discretized by the following formula:

$$sim_{ij} := \begin{cases} 0, & \text{if } sim_{ij} < \epsilon_{sim} \\ 1, & \text{otherwise} \end{cases} . \tag{5.7}$$

By changing the weight of an edge to zero the graph is implicitly pruned since zero-weighted edges are discarded. Changing the weight to a value of one is optional and not necessarily needed, but

[4] An implementation of Chinese Whispers can be found at http://wortschatz.informatik.uni-leipzig.de/~cbiemann/software/CW.html

[5] The Small World property for graphs says that a vertex can be reached from any other vertex within a certain number of hops (Milgram 1967).

gave better results in preliminary experiments. An illustration of the graph clustering procedure can be found in Figure 5.2.

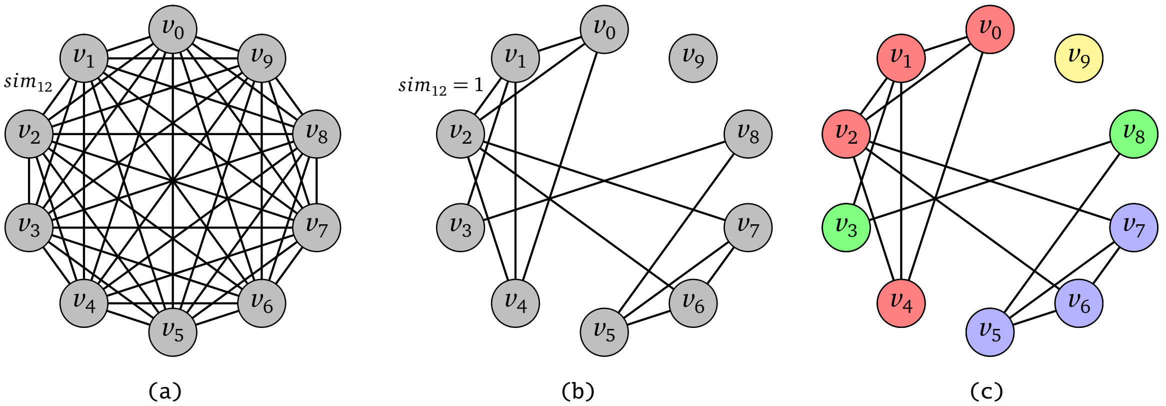


Figure 5.2.: *Graph clustering illustration. (a) shows the complete graph after computing similarity between every pair of vertices. Here the edges are still weighted. (b) shows the graph after discretizing the weight of all edges. Edges with weight zero are discarded. (c) shows a possible result after running ChineseWhispers. Different colors for vertices denote different assigned class labels. Hence, vertices with the same color form the desired clusters and are assumed to be related.*

In particular, the CW algorithm is equipped with a parameter that defines various options for determining the predominant class in the neighborhood of a certain vertex (cf. Listing 5.2:11). The neighborhood of a vertex v_i is defined as the set of vertices that are connected via an edge to v_i. In the following, $n(v_i)$ will be used as a method for retrieving the neighbors of a node v_i. Each class label is assigned a certain weight which is computed by considering vertices and class labels of vertices in the neighborhood of v_i. The *class label weighting scheme* controls the computation of the weight of a certain class label and thus the assignment of a class to v_i in the update step of CW. Let $w_k^{(i)}$ be the weight of a class with label k for the vertex v_i, then c_i is assigned the label with the maximum weight according to $\arg\max_k(w_k^{(i)})$. Biemann (2012) presented various strategies for computing the label weight $w_k^{(i)}$ which are summarized in Table 5.3.

See Biemann (2012, pp. 94–103) for a detailed description of the weighting schemes as well as the impact on the resulting class label for v_i and hence on the final clustering.

The final chaining procedure is then straightforward with analogies to the LDA-MM approach: The LDA-GM algorithm assigns every candidate lexical item w_i of a certain document d which is assigned the same class label c_i to the same chain.

Table 5.3.: *Chinese Whisper's class label weighting schemes for controlling the new class assignment of a vertex v_i. Note that the definitions have been adapted for the current graph where we have only edge weights with a value of one.*

name	formula
top	$w_k^{(i)} = \sum_{\substack{v_j \in n(v_i) \\ c_j = k}} 1$
dist	$w_k^{(i)} = \sum_{\substack{v_j \in n(v_i) \\ c_j = k}} \frac{1}{\lvert n(v_j) \rvert}$
dist-log	$w_k^{(i)} = \sum_{\substack{v_j \in n(v_i) \\ c_j = k}} \frac{1}{\log(1 + \lvert n(v_j) \rvert)}$

Level Two Link Extension

Level two links can be extracted using a property of the CW algorithm. CW computes the predominant class label in the neighborhood of a particular vertex with an argmax function. Here we can assess by considering also the second dominating class in the neighborhood. The last iteration of the CW algorithm then delivers not only the final class label for a node, but also an alternative. This information is then used for linking analogously to the linking procedure in the LDA-MM approach.

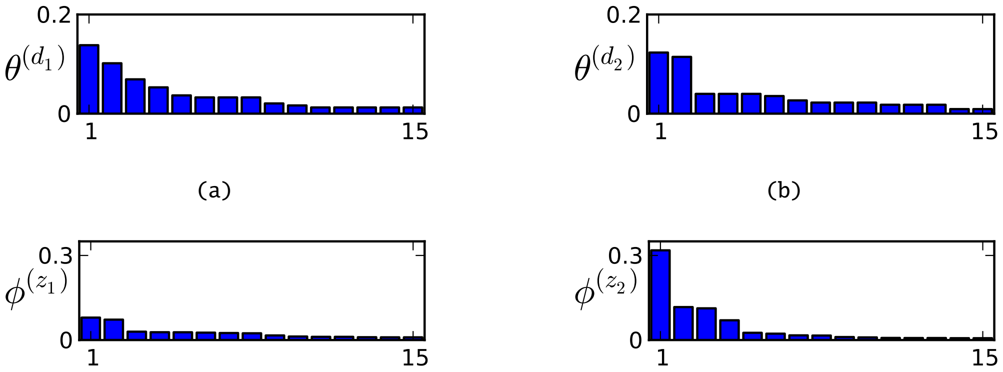
5.2.3 LDA Top-N Method (LDA-TM)

The LDA-TM methodology is different to the methodologies mentioned before in that is uses the information of the per-topic word distribution $\phi^{(z)} = P(w|z)$ and the per-document topic distribution $\theta^{(d)} = P(z|d)$. Given a parameter n referring to the top n topics to choose from $\theta^{(d)}$ and a parameter m referring to the top m words to choose from $\phi^{(z)}$ the main procedure can be described as follows:

1. for $z \in$ top n topics in $\theta^{(d)}$
    a) chain the top m words in $\phi^{(z)}$.

As it turns out in some preliminary experiments, it is not viable to assign a global value for n and m. In some documents a lot of topics are involved and in some documents only a few topics are involved. This observation is similar for topics: Some topics are influenced by a lot of words and some topics are influenced by only a few words. Thus we can not assume that a certain value of n and m that produce good results for one document will produce good results for another document.

As an illustration consider Figure 5.3 which shows sorted $\theta^{(d)}$ and sorted $\phi^{(z)}$ probabilities for different documents, resp. topics. Here we can see the actual values of the top15 topics of different documents as well as the values of the top 15 words for different topics which emphasizes the problem of choosing good n and m values for all documents equally. Mind that the distributions obviously depend on the chosen hyperparameters α and β. Though, the values chosen here have proven to give good results in the experiments described in Chapter 6 as they are from the final model. Also, by fixing n and m globally, we would limit the number of chains as well as the number of chain members for all documents equally. For these reasons, drawbacks of the method became clear.



(a)

(b)

(c)

(d)

Figure 5.3.: *Plots of selected per topic word distributions ϕ in (a,b) and per word topic distributions θ in (c,d). The probability distributions are sorted by their value and pruned after the top 15 topics / words. The model is estimated with $T = 500$, $\alpha = \frac{50}{K}$, $\beta = 0.001$ on a dataset comprising $\sim$13K documents with a vocabulary of $\sim$100K words. With respect to (a) and (b) it is hard to decide where to set the boundary for the top n topics. The same problem holds for the top m words of a topic regarding (c) and (d).*

It is left for future investigations if good n and m can be guessed somehow, e.g. by document length or type-token ratios or the like. At first sight, they do not depend on document length and for the time being a heuristic is applied which bounds the initial values n and m by a threshold ϵ_n and ϵ_m of the respective probability values. Listing 5.3 shows the procedure more comprehensively.

Despite the fact that ϕ and θ are computed by considering only the last assignment of the Gibbs sampling algorithm, the top n topics and top m words are somewhat stable. This is kind of natural, since the proportions of the prominent topics and words do not change heavily after a certain number of sampling iterations.

Note that although the number of chains and chain members for each chain is bound and one could initially think of different documents with same chain structure, in practice the number of gener-

Listing 5.3: *LDA-TM procedure.*

```
 1 $chainset \leftarrow \emptyset$
 2 $\vartheta \leftarrow \text{sort\_descending}(\theta^{(d)})$
 3 for $i \in [1, n]$ do
 4     $z \leftarrow \vartheta_i$
 5     $\varphi \leftarrow \text{sort\_descending}(\phi^{(z)})$
 6     $chain \leftarrow \emptyset$
 7     for $j \in [1, m]$ do
 8         $w \leftarrow \varphi_j$
 9         if $w \in d \wedge \varphi_j > \epsilon_\phi \wedge \vartheta_i > \epsilon_\theta$ then
10             $chain \leftarrow chain \cap w$
11         end if
12     end for
13     if $|chain| > 0$ then
14         $chainset \leftarrow chainset \cap chain$
15     end if
16 end for
```

ated chains as well as the number of chain members varies a lot between individual documents. Often some of the top m words for a topic do not even occur in a particular document.

Blei and Lafferty (2009a) proposed a simple word reordering technique that down weights words that have a high probability value in all topics which they use preferably instead of the word probability $\phi^{(z)}$. Line 5 in Listing 5.3 then changes to "$\varphi \leftarrow \text{sort\_descending}(score^{(z)})$" with

$$score_w^{(z)} = \phi_w^{(z)} \log \left(\frac{\phi_w^{(z)}}{\sqrt[K]{\prod_{i=1}^{K} \phi_w^{(i)}}} \right) . \quad (5.8)$$

Preliminary experiments indicated a non-significant performance gain and hence this technique is not further considered in the final evaluation.

Level Two Link Extension

Level two links are created by computing the similarity between every pair of the top n topic distributions. As a measure of similarity the cosine distance is chosen. For each pair of topics whose similarity is above a certain threshold a link is created between a random lexical item of each corresponding lexical chain.

5.3 Term Co-Occurrence Significance

Statistical natural language processing typically deals with the computation of occurrence and co-occurrence events. TMs for example are nothing else but co-occurrence methods for words and topics and topics and documents computed from the co-occurrence of words in documents (cf. the section above). In this section, another type of co-occurrence will be studied namely the co-occurrence of a word with another word.

Word co-occurrences are typically expressed by statistical events such as the presence and the absence of a certain word in given window. Here, the window is a sentence and the presence of a word w_A in a certain sentence as well as the presence of another word w_B in that sentence is expressed by the events A and B respectively. Measuring the significance between the events A and B— i.e. putting the impact of the events on each other into numbers — is typically achieved with the help of contingency tables. A contingency table presents the quantitative information obtained from raw data collections of two or more events in a clean concise way. Table 5.4 shows the structure of a 2×2 contingency table with n being the number of samples — in our case the number of sentences — in the analyzed data collection and n_{XY} being the number of samples in that the events X and Y occurred. Mind that X is either A or $\overline{A}$ (pron. "not A") which denote the presence or absence of w_A in a sentence and Y is either B or $\overline{B}$ denoting the presence or absence of w_B in that sentence. The marginal sums describe the number of samples in that the individual events occurred or not occurred, e.g. the word w_A occurred in n_A sentences and did not occur in $n_{\overline{A}}$ sentences; a nice property that implies only the requirement of n, n_A, n_B and n_{AB} from which the other quantities can be easily computed.

Table 5.4.: *A contingency table describing the events A and B quantitatively.*

	A	$\overline{A}$	$\sum$
B	n_{AB}	$n_{\overline{A}B}$	n_B
$\overline{B}$	$n_{A\overline{B}}$	$n_{\overline{A}\overline{B}}$	$n_{\overline{B}}$
$\sum$	n_A	$n_{\overline{A}}$	n

Using a contingency table a lot of statistical measures respectively likelihoods can be directly computed e.g. the odds-ratio, Pearson's correlation coefficient, Yule's Q, Yule's Y, Pearson's χ^2, mutual information, log likelihood ratio, z-score, t-score, multinomial-likelihood, binomial-likelihood, Poisson-likelihood, and many more. See e.g. Evert (2005)[6] for a detailed list of significance measures.

[6] http://www.collocations.de/AM/

In this thesis the log likelihood ratio (LLR) (Dunning 1993) is used for testing the significance of the events A and B. The LLR as a statistical significance measure is especially used in language data because of its adaptability for rare events. Recall that according to Zipf's Law, words with very low frequency in everyday use make up the bulk of the vocabulary of a language. Also, these rarely occurring words include many meaning bearing words. Another advantage of the LLR is the asymptotic behavior to the χ^2 statistics which allows for the critical values rejecting the null hypothesis under the χ^2 distribution to be valid also for LLR. The results are equivalently interpretable to the results of the χ^2 test, i.e. the null hypothesis states that the two events A and B are independent and a higher LLR value rejects the null hypothesis with a higher confidence. The log likelihood ratio can be computed with the help of a contingency table by the following formula:

$$\begin{aligned} LLR &= -2\log(\lambda) \\ &= 2 \times [n\log(n) - n_A\log(n_A) - n_B\log(n_B) + n_{AB}\log(n_{AB}) \\ &\qquad + n_{\overline{AB}}\log(n_{\overline{AB}}) + n_{\bar{A}B}\log(n_{\bar{A}B}) + n_{A\bar{B}}\log(n_{A\bar{B}}) \\ &\qquad - n_{\bar{A}}\log(n_{\bar{A}}) - n_{\bar{B}}\log(n_{\bar{B}})] \end{aligned} \tag{5.9}$$

See Bordag (2007) for details. In practice, the LLR between each pair of words in a given corpus is computed using the TinyCC[7] program (Quasthoff et al. 2006) which is able to handle the co-occurrences of a huge number of words.

The lexical chaining procedure is then straightforward with analogies to the LDA-GM procedure. First the LLR scores for a big data collection are computed. When processing a certain document for lexical chaining the scores are retrieved for every word pair in that document and inserted into an adjacency matrix as shown in Table 5.2. Because of the symmetry property of the log likelihood ratio only the upper or lower triangular matrix needs to be created. From this matrix an undirected graph as shown in Figure 5.2a can be created and pruned with a specified threshold ϵ_{llr} continued by the clustering of the pruned graph with Chinese Whispers. Each word with the same assigned class label is then assigned to the same chain. This procedure is exactly the same as the "second part" of the LDA-GM procedure whereby only the similarity scores between the per-word topic distributions provided by LDA are exchanged by the LLR scores. This methodology is simply called the LLR Graph Method (LLR-GM).

Level Two Link Extension

Level two links are created according to the LDA-GM approach, i.e. by remembering the second dominant class label in the neighborhood of a certain vertex and using the information for tying

[7] http://wortschatz.uni-leipzig.de/~cbiemann/software/TinyCC2.html

the item together with a random item that is assigned this class label (cf. the linking procedure of LDA-GM in Section 5.2.2).

5.4 Final Notes

All methodologies explained above rely on the statistical analysis of big data collections which is called the *training set* or *training documents* henceforth. As this, they can be directly applied to any document that is in the training set after the analysis.

A plus of the chosen methods is that they can also be applied to new unseen documents that are not in the training set, called *test documents* henceforth. It is possible to create a *trained model* based on the training set and to use this as background information for chaining the test documents.

In order to be adequately applicable, the test documents should be of the same domain as the training documents. This is a kind-of obvious restriction since we can not expect the methods to work well when comparing apples and pears.

Though, it may happen that the training documents consist of a slightly different vocabulary as the test documents. In such cases unknown words cannot be chained adequately. A simple post processing step is thus performed after the chaining process of the individual methods which is the chaining of repetitions of unknown words. Recall that repetition is the simplest form of a lexical cohesive relation.

6 Empirical Analysis

In this chapter lexical chains produced by the algorithms developed in Chapter 5 are compared to the manually annotated lexical chains developed in Chapter 4 using the comparison measure developed in Chapter 3.

The lack of manually annotated datasets led to the fact that most of the previous lexical chaining approaches are evaluated extrinsically by measuring the performance of a particular task that uses lexical chain information. This is a valid decision when no data is available but one must not forget the reason why no such data exists.

The subjective interpretation of an annotator has an immense impact on the structure of the resulting lexical chains. Interpreting a chaining algorithm as an annotator and measuring the quality of lexical chains extrinsically based on a certain task implies almost always a subjected evaluation to that task. This is perfectly valid since in essence it is the result of the actual task that finally counts, but one has to be aware that it is actually not the quality of lexical chains in general that is measured rather the quality of lexical chains with respect to the particular task is evaluated.

This thesis uses a fraction of the annotated lexical chains developed in Chapter 4 for the intrinsic evaluation of the various statistical algorithms. The other fraction of the annotations is used for the determination of the algorithm's parameters; a procedure called *model selection*. Additionally, various trivial baseline algorithms as well as two knowledge-based algorithms are evaluated whose results are then compared to the results of the annotators and the results of the developed algorithms.

6.1 Corpus

The test corpus consists of 100 selected documents of the Salsa/Tiger Corpus which are annotated during the work of this thesis as explained in Chapter 4.

The training corpus consists of the Salsa/Tiger documents minus the test documents with inappropriate texts being heuristically identified and discarded. Precisely, the Salsa/Tiger Corpus consists of news text from the German newspaper “Frankfurter Rundschau” around 1992. Some of the texts are just short summaries about various happenings which are a kind of news ticker texts. Most of these texts share the same structure and are heuristically identified by e.g. the title which is often “Kurznachrichten” or “Nachrichten-Börse” or the like others are signed by multiple press agencies (dpa, rtr, etc.) in different paragraphs which is a clear indicator of false document boundaries. These documents are discarded since they would merely confuse the statistical methods. Additionally the training corpus is enriched with news texts from the same newspaper around 1997+ with equal structure. Further, short texts with less than ten candidate lexical items are discarded. The individual sizes of the corpora and the sum as the size of the used training corpus are listed in Table 6.1.

Table 6.1.: *Size of the training corpus.*

corpus	#usable documents
Salsa/Tiger	$1,211$
FR	$12,264$
$\sum$	13,457

6.2 Experimental Setup

In order to qualitatively compare the output of the lexical chaining approaches some trivial baseline algorithms as well as two non-trivial knowledge based algorithms are considered for comparison. The baseline algorithms are:

All In One: Each candidate item is assigned to the same lexical chain which results in a single lexical chain per document; no links are created.

Repetitions: Each candidate item and its lexical repetition is assigned a single lexical chain which results — in case of no repetitions — in as many lexical chains as there are candidate items; no links are created.

Random: Candidate lexical items are randomly tied together to form sets of lexical chains. Level two links are then created analogously between the resulting lexical chains. Two parameters control the amounts of cohesive ties and level two links in order to build in essence a similar structure as the manually annotated lexical chains.

S&M GermaNet: Silber and McCoy's (2002) algorithm (cf. Section 2.2) in combination with GermaNet[1] (Hamp and Feldweg 1997) is used. The algorithm was implemented in the context of the work by Jakob (2007) and revised in the context of the work by Schwager (2008).

[1] GermaNet is the german equivalent of WordNet®.

G&M GermaNet: Galley and McKeown's (2003) algorithm aiming for word sense disambiguation (cf. Section 2.2) also in combination with GermaNet and implementation by Jakob and Schwager (see S&M GermaNet above).

The general evaluation procedure using a set of annotated documents as test set is as follows:

1. Randomly split the test set into two equally sized test sets.
2. Use one test set for model selection, i.e. gather the parameters for the individual methods that maximize the evaluation measure.
3. Use the other test set and the parameter set determined by the model selection step for the evaluation of the various approaches as well as the baseline methods.

Input to the LDA methods are verbs, nouns and adjectives in their lemmatized form as well as word combinations as described in Section 5.1 also in their lemmatized form. A list of stop words was maintained in order to pre-filter uninformative words. Additionally, words that occur in more than 1⁄3 of the training documents are considered high frequency words and are discarded as well. Words that occur in less than two documents are also discarded since these do not give any incorporating information. The size of the vocabulary thus dropped from a number of around 325K words to 100K words. This strategy improves the results of the topics estimated by LDA which is the foundation of the lexical chaining algorithms, and decreases the complexity for the computation of both, the LDA topics as well as the log likelihood ratios. In fact, the LLR method needs as input only the candidate lexical items from the test set, as the graph is built exclusively of these.

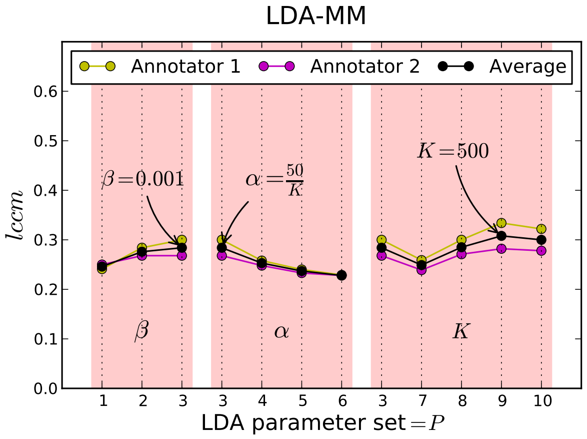
6.3 Model Selection

Model selection is the process of empirically testing various parameter sets for the different algorithms. The parameters for the different methods are listed in Table 6.2

Table 6.2.: *Model parameters for the individual statistical methods.*

Method	Parameters	
LDA-MM	$\{K, \beta, \alpha, iters\} \cup$	$\{\}$
LDA-GM		$\{similarityfunction,\ labelweightscheme, \epsilon_{sim}\}$
LDA-TM		$\{n, m, \epsilon_{\theta}, \epsilon_{\phi}\}$
LLR-GM	$\{labelweightfunction, \epsilon_{llr}\}$	

The parameters $\{K, \beta, \alpha, iters\}$ are the parameters for the LDA algorithm. Because estimating an LDA model is quite time-consuming and the parameter space is quite large for certain methods, only one LDA model is estimated and is used for the three LDA-* methods. The model is estimated using the LDA-MM approach since no other parameters are needed here. Although the model selection complexity dropped a great amount it is still too complex to validate all four LDA parameters. Hence, the parameters are determined heuristically one after another starting with the default parameters recommended in the literature e.g. Blei et al. (2003). Starting with $K = 100$, $\alpha = {}^{50}/_{K}$, $\beta = 0.1$, and $iters = 10,000$ the parameters are estimated in three rounds. Note that the $iters$ parameter is determined with the first evaluated model by evaluating a snapshot of the model taken after every 500 iterations. After 1000 iterations the values have not improved much, so that 1000 iterations suffice to get a good model. The other parameters were then changed one after the other and the best parameter was chosen. Figure 6.1 shows the development of the best estimated model.

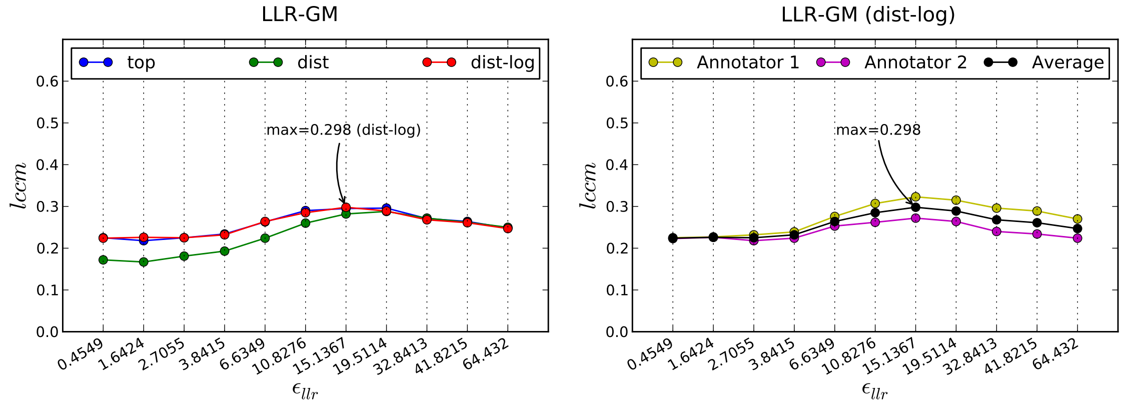


P	$iters$	β	α	K
1	1,000	0.1	${}^{50}/_{K}$	100
2	⋯	0.01	⋯	⋯
3	⋯	0.001	⋯	⋯
4	⋯	⋯	${}^{5}/_{K}$	⋯
5	⋯	⋯	${}^{0.5}/_{K}$	⋯
6	⋯	⋯	${}^{0.05}/_{K}$	⋯
7	⋯	⋯	${}^{50}/_{K}$	50
8	⋯	⋯	⋯	250
9	1,000	0.001	${}^{50}/_{K}$	500
10	⋯	⋯	⋯	1000

Figure 6.1.: *Model selection of the time-consuming LDA parameters. The ninth parameter set maximizes the average of the comparison measure of both annotators.*

This greedy strategy reduces the parameter space to be adequately processable but finds most probably just a local optimum. When time is of minor importance and a faster LDA algorithm like *online learning* (M. D. Hoffman et al. 2010) is used, the results will most likely improve further.

For the search of the other optimal parameter values, the full parameter space can be regarded. Figure 6.2 shows exemplarily how the values for the LLR-GM parameters are determined. The average of both annotators is decisive for the determination of the parameter values. In particular, as Figure 6.2 shows, the *top* label weighting scheme and the *dist-log* label weighting scheme deliver almost the same results.



(a)

(b)

Figure 6.2.: *Selecting the best LLR-GM parameters. The values of the left plot are based on the average of both annotators and shows for each label weighting scheme and threshold value ϵ_{llr} the $lccm$ score. The LLR-GM approach using the* dist-log *label weighting scheme and a pruning threshold of* ∼15 *finally maximizes the $lccm$ score for the model selection test set. The right plot shows in the $lccm$ scores for different ϵ_{llr} using the* dist-log *scheme for both annotators.*

The other parameters are determined analogously and the final parameter values are summarized in Table 6.3.

6.4 Evaluation

Using the final parameter values determined in the model selection phase, the methods can be evaluated against the trivial baseline methods as well as the knowledge-based algorithms by Silber and McCoy (2002) and Galley and McKeown (2003) using GermaNet as knowledge base each. For fairness purposes, because the knowledge based baselines do not handle well multi-word expressions, lexical items that span over multiple words are mapped to its rightmost term which is assumed to be the head of the compound or the main message of the word combination, e.g. “dirty money” is mapped to “money”. Additionally, singleton chains, i.e. chains that contain only a single lexical item are omitted unless the respective lexical item is not linked by a level two link. This allows better comparison between the approaches.

The results are shown in Figure 6.3. Here, the baseline scores are averaged between the two annotators. Note that all statistical approaches developed here beat the knowledge-based baseline methods by a significant amount. Furthermore, the All-In-One baseline method as well as the Repetition baseline method have relatively high scores because of a caveat of the comparison measure. Although averaged by the ARI measure it is still influenced by chance — recall that the

Table 6.3.: *Final parameter values.*

Method	Parameter
LDA-MM	$K = 500$ $\beta = 0.001$ $\alpha = \frac{50}{K}$ $iters = 1000$
LDA-GM	$K, \alpha, \beta, iters$ $similarityfunction$ = cosine similarity $labelweightscheme$ = dist-log ϵ_{sim} = 0.95
LDA-TM	$K, \alpha, \beta, iters$ n = 10 m = 20 ϵ_{θ} = 0.2 ϵ_{ϕ} = 0.2
LLR-GM	$labelweightscheme$ = dist-log ϵ_{llr} = 15.1367

ARI measure is actually chance corrected and would give the All-In-One baseline a zero score as well as the Repetition baseline a lower score. Despite the fact that the trivial baselines are indeed undesired lexical chain structures — consider that these are not helpful in any practical application relying on lexical chains — we do not want to ignore the chances of a chain containing all elements or the chances of multiple chains each containing a single item which are basically still valid just undesired structures.

The GermaNet baselines perform nearly the same and both perform slightly worse than the Repetition baseline which simply originates from the fact that GermaNet contains only a fraction of the lexical items chained by the annotators. Recap that the $lccm$ measure penalizes both, not considering a lexical item at all and chaining a lexical item into a “wrong” chain. Since the trivial baselines build only “wrong” chains but consider at least all wanted lexical items — in fact they consider also undesired lexical items but the number of these is much smaller than the number of useful candidate items — they are not punished that harshly.

Level 2 Links

Currently, the model selection as well as the final evaluation offer only the results for lexical chains ignoring the level two links. As argued in the annotation scheme in Chapter 4 these links bear

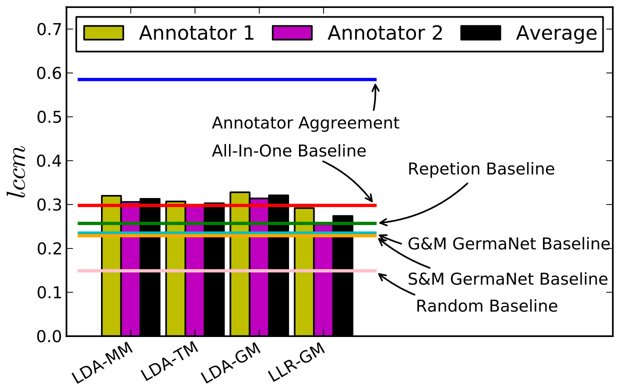


Figure 6.3.: *The final evaluation results comparing only chains.*

important information and are crucial for the lexical cohesive structure of a text. One approach to evaluate this information is to merge those chains that contain lexical items that are tied by a level two link which is also applied here.

Figure 6.4 shows the evaluation results of the merged chains. Note that the annotator agreement slightly dropped because of the caveat of the $lccm$ mentioned above. Because of the merging of chains via links, a text now contains fewer chains with more lexical items each, which is more similar to that chain structure built by the All-In-One baseline which is hard to beat here. Figure 6.5 illustrates this fact on an example text. Mind that none of the baseline methods is able to build level two links except the random baseline. The difference in the score of the respective baseline method originates entirely from the merger of the annotators chains. Despite all this we can clearly say that the statistical approaches perform much better than the knowledge based approaches.

Final Notes

Table 6.4 shows some quantitative numbers of the extracted lexical chains made by the different statistical algorithms as well as the knowledge based baseline algorithms. The quantitative numbers based on lexical chains made by the annotators can be looked up in Table 4.1 for the purpose of comparison. Although the evaluation set is just 50% of the complete test set, the relative numbers are nearly equal since the documents are independently randomly distributed in the model selection- and evaluation test sets.

As the numbers reveal, the structure of lexical chains extracted by the individual methods are somewhat different, at least in average. E.g. the LDA-MM approach chains and links a lot more items than the other statistical methods. This comes from the fact that it simply creates a lot more links between items that would otherwise be removed from consideration because they form

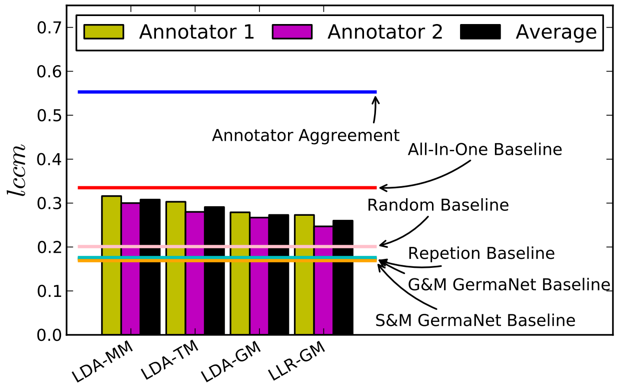


Figure 6.4.: *The final evaluation results comparing merged chains.*

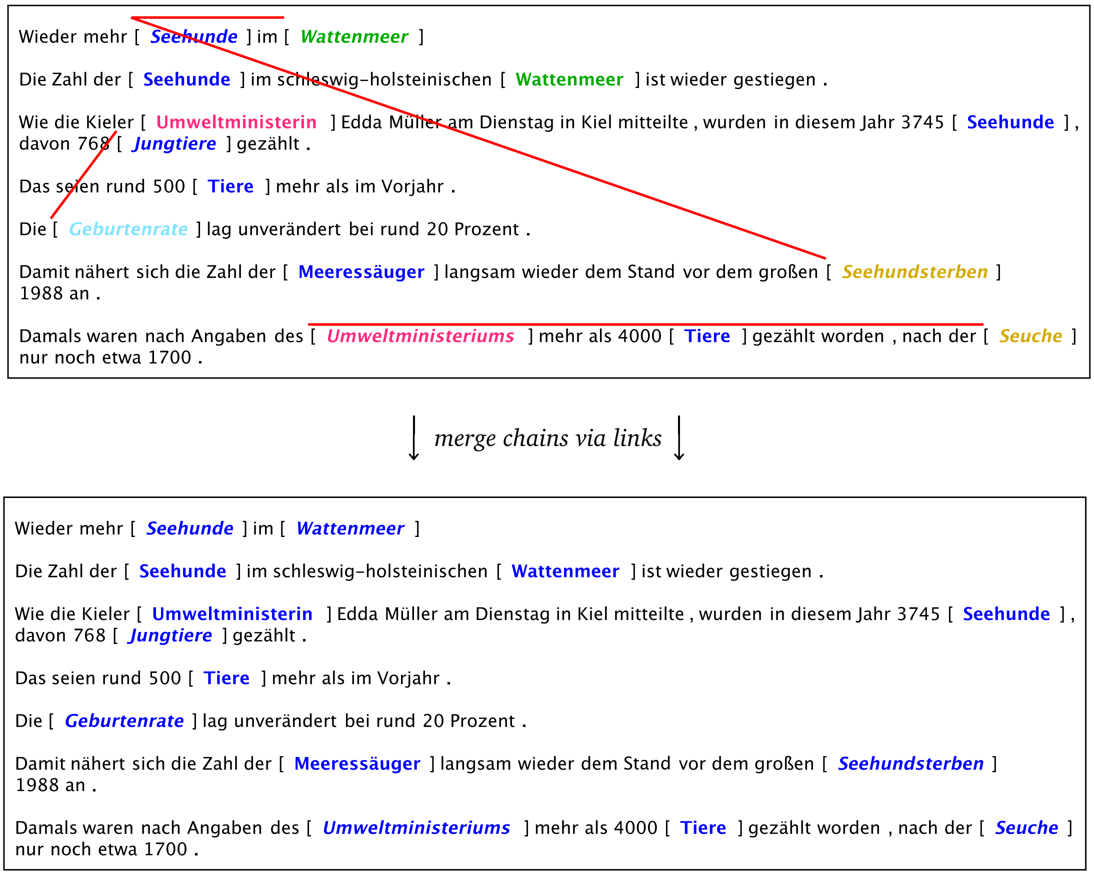
Table 6.4.: *Automatically extracted lexical chains in numbers. In average a document contains 51.58 candidate lexical items.*

	LDA-MM	LDA-GM	LDA-TM	LLR-GM	S&M	G&M
avg. number of lexical items per doc.	38.20	29.32	30.82	24.74	14.40	15.29
avg. number of chains per doc.	13.80	9.12	7.32	7.76	5.83	5.71
avg. number of links per doc.	8.60	2.06	1.44	1.84	–	–
avg. size lexical chains	2.82	3.41	4.61	3.36	2.48	2.68
avg. number of merged lexical chains	5.76	7.06	5.98	5.94	–	–
avg. size merged lexical chains	8.29	4.45	5.57	4.96	–	–

singleton chains, i.e. chains that contain only one lexical item. Since the items are linked by level two links their corresponding singleton chain is kept.

The graph methods (LDA-GM, LLR-GM) as well as the top-n method (LDA-TM) on the other hand perform an implicit filtering on the candidate lexical items by being stricter when creating level two links. These methods create in average a lot less level two links and they also create larger lexical chains thus creating broader concepts in advance. But keep in mind that the number of links also influences the number of chains when links occur between singleton chains.

The knowledge based algorithms by Silber and McCoy and Galley and McKeown extract fewer chains than the statistical approaches and consider in average less lexical items as related. This is based on the fact that candidate lexical items are often not contained in the GermaNet lexicon.



Wieder mehr [***Seehunde***] im [***Wattenmeer***]

Die Zahl der [**Seehunde**] im schleswig–holsteinischen [**Wattenmeer**] ist wieder gestiegen .

Wie die Kieler [**Umweltministerin**] Edda Müller am Dienstag in Kiel mitteilte , wurden in diesem Jahr 3745 [**Seehunde**] , davon 768 [***Jungtiere***] gezählt .

Das seien rund 500 [**Tiere**] mehr als im Vorjahr .

Die [***Geburtenrate***] lag unverändert bei rund 20 Prozent .

Damit nähert sich die Zahl der [**Meeressäuger**] langsam wieder dem Stand vor dem großen [***Seehundsterben***] 1988 an .

Damals waren nach Angaben des [***Umweltministeriums***] mehr als 4000 [**Tiere**] gezählt worden , nach der [***Seuche***] nur noch etwa 1700 .

Wieder mehr [***Seehunde***] im [***Wattenmeer***]

Die Zahl der [**Seehunde**] im schleswig–holsteinischen [**Wattenmeer**] ist wieder gestiegen .

Wie die Kieler [**Umweltministerin**] Edda Müller am Dienstag in Kiel mitteilte , wurden in diesem Jahr 3745 [**Seehunde**] , davon 768 [***Jungtiere***] gezählt .

Das seien rund 500 [**Tiere**] mehr als im Vorjahr .

Die [***Geburtenrate***] lag unverändert bei rund 20 Prozent .

Damit nähert sich die Zahl der [**Meeressäuger**] langsam wieder dem Stand vor dem großen [***Seehundsterben***] 1988 an .

Damals waren nach Angaben des [***Umweltministeriums***] mehr als 4000 [**Tiere**] gezählt worden , nach der [***Seuche***] nur noch etwa 1700 .

Figure 6.5.: *Manually annotated document. Words belonging to the same chain are colored the same. Red lines illustrate level two links between the involved items. The upper text shows the originally annotated document and the lower text shows the chain structure after the merging operation. Five chains and four links are present before the merger and only one chain for the whole text exists after the merger.*

7 Summary & Conclusion

7.1 Conclusion

Validity of the Hypothesis

The central theme of this thesis is the question if statistical methods can compare with knowledge resource based methods for the task of lexical chain extraction. In order to test this hypothesis, four newly developed statistical methodologies for lexical chain extraction have been evaluated against some trivial chaining techniques as well as some established knowledge based chaining techniques with the result that statistical methods significantly outperformed the knowledge based approaches. Hence, the outcome of the evaluation confirms the hypothesis.

As argued in the problem statement (cf. Sec. 1.1), the performance of the knowledge based approaches highly depends on the quantity and quality of the content of the utilized knowledge resource. GermaNet as the knowledge resource used throughout this thesis, though quite comprehensive, is not comparable in quantity and quality to its English counterpart WordNet. Thus a quantitatively larger and a qualitatively better knowledge resource will most probably improve the performance of the knowledge based chaining approaches.

Nevertheless, comparable statistical approaches are always preferable over knowledge based approaches as the costs for building a statistical model are much cheaper than the cost for creating a reliable knowledge resource.

Limitations & Suggestions

What is a strength of statistical methods is also its major limitation namely the reliance on good data collections. This means especially that a corpus of an adequate size — larger is always better here — and also of good and uniform quality lets the statistics show its full strength. What is needed is a hand for the preprocessing of the data as well as a good feeling for the parameters of the statistical models especially the LDA methods.

A major problem of the algorithms is the choice of good candidate lexical items. This is also a problem of knowledge based approaches but of minor interest here since a common approach is to assume all items contained in a knowledge resource to be relevant at all. During this thesis, a simple heuristic is applied for the detection of candidate lexical items. More sophisticated approaches could help here, e.g. the identification of compounds like “small talk” or the like. The statistical methods could also gain from prior compound splitting like “bedroom” into “bed” and “room” as these typically encode multiple topics. This may not be a big problem for the English language but applies particularly for the German language where words can be compounded excessively e.g. the word “Donaudampfschiffahrtsgesellschaftskapitän”[1] is a compound of various individual words. This preprocessing step could improve the statistical reasoning and thus the final statistical model.

Also, in the context of this thesis the algorithms as well as the manual annotations were mostly limited to nouns. This decision was met in order to limit the workload and perform a fair evaluation — recall that knowledge based approaches only consider nouns due to structural limitations — and is not a general limitation. The developed methods are applicable to any kind of words. To include verbs and other parts-of-speech in lexical chaining — just as theoretically defined — will most likely enhance the intuitive quality of the lexical chains themselves as well as most natural language processing tasks that rely on lexical chains, e.g. summarization or text segmentation or the like.

7.2 Future Divisions & Applications

Carrying out an enhanced annotation project that involves maybe hundreds of annotators could give insights into the diverse interpretations of a text by different readers. It would be an interesting investigation if the interpretations follow any specific rules. With that many annotators maybe an “average interpretation” could be generated which can then be used for further processing.

Another direction is the development of an ultimate lexical chain comparison measure. The here developed $lccm$ is a valid choice but it is still biased by a tradeoff (cf. Sec. 3.4). Fortunately the clustering comparison domain is under heavy development so that better clustering comparison measures may be developed which qualify as lexical chain comparison measure.

Further directions include the development and evaluation of natural language processing applications that rely on statistically extracted lexical chains. Applications that already benefit from lexical chain extraction are manifold. These need to be adapted to the new methods. An advantage is here that next to the lexical chains themselves, also the internal structure of the developed techniques can be exploited — the similarity graphs of the both graph based approaches (LDA-GM and LLR-GM) for example.

Also, more sophisticated methodologies can be developed that combine statistical and knowledge based approaches which can then even be extended to supervised approaches that learn from the annotated data.

[1] “captain of the Danube Steam Shipping Company”

7.3 Summary

Summarizing, in this thesis statistical methods were examined for their applicability for the identification of lexical chains in a cohesive text.

Four algorithms were developed that exploit the statistical foundation of these methods in different ways in order to extract lexical chain information. The algorithms were built on top of two branches from statistical natural language processing which are (a) probabilistic topic models, *latent Dirichlet allocation* (LDA) in particular, and (b) statistically significant word co-occurrences. These algorithms have been extended in order to identify not only the cohesive structure of lexical chains, but also the structure of chain interaction defined by cohesive harmony which is instantiated through so-called level two links here.

In order to measure the performance of the approaches an extensive survey of suitable evaluation measures was performed. This survey introduced desirable characteristics for the problem of comparing different sets of lexical chains and examines various well-known measures from the clustering domain. Finally a combination of the *adjusted Rand index* and the *normalized basic merge distance* (NBMD) was chosen which best reflects most of the desired characteristics. The developed measure is simply called the $lccm$ (lexical chain comparison measure) and was used for the analysis of the interannotator-agreement in the performed annotation project as well as for the model-selection and final evaluation of the lexical chain extraction algorithms.

An annotation project was realized in order to develop a dataset building a reliable basis for the general evaluation of lexical chains which includes also the level two links. The annotation project was kept extensible and can be carried on due to the provision of annotation guidelines and the adaptation of a suited computer-aided annotation framework. During this thesis, two annotators annotated around one hundred documents from the German SALSA 2.0 / Tiger 2.1 corpus, thereby enhancing the already comprehensively annotated corpus with information about lexical cohesion. The annotated texts have been analyzed with regard to their inter-annotator agreement using the $lccm$. The analysis has shown that high inter-annotator agreement is merely possible due to the general problem of subjective interpretations of texts by individual readers.

Using the annotated data and the proposed comparison measure the performance of the developed statistical methods is measured and validated against various trivial lexical chain extraction strategies as well as some established knowledge based lexical chain extraction algorithms. Opposing to previous approaches, the evaluation was performed entirely intrinsic which means that the extracted lexical chains were compared directly to the manually annotated lexical chains. This strategy is completely new since previous approaches measured the performance of the algorithms solely on the task the lexical chains were intentionally used for.

The results of the evaluation have shown that lexical chaining algorithms based on statistical methods significantly outperform lexical chaining algorithms based on knowledge resources. Hence, it can be said that statistical methods qualify for the extraction of lexical chains in a text and are a preferable choice over knowledge resource based algorithms especially in uncommon domains or resource-scarce languages.

List of Figures

List of Listings

List of Tables

Bibliography

Ajmera, J., Bourlard, H., and Lapidot, I. 2002. Unknown-Multiple Speaker clustering using HMM. In *Proceedings of the international conference of spoken language processing.* ICSLP '02. Denver, Colorado, USA.

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval* 12 (4): 461–486.

Bagga, A. and Baldwin, B. 1998. Algorithms for Scoring Coreference Chains. In *Proceedings of the linguistic coreference workshop at the first international conference on language resources and evaluation,* 563–566. LREC '98. Granada, Spain.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. 1998. The Berkeley FrameNet Project. In *COLING '98: Proceedings of the 17th International Conference on Computational Linguistics,* 86–90. Vol. 1. Montreal, Quebec, Canada.

Barzilay, R. 1997. Lexical Chains for Summarization. Master's thesis, Ben-Gurion University of the Negev.

Barzilay, R. and Elhadad, M. 1997. Using Lexical Chains for Text Summarization. In *Proceedings of the ACL workshop on intelligent scalable text summarization,* 10–17. Madrid, Spain.

Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S., and Widom, J. 2009. Swoosh: a generic approach to entity resolution. *The VLDB Journal* 18:255–276.

Biemann, C. 2006. Chinese Whispers – an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing. In *Proceedings of textgraphs: the second workshop on graph based methods for natural language processing,* 73–80. New York City, USA.

———. 2012. *Structure Discovery in Natural Language.* Theory and Applications of Natural Language Processing. Springer Berlin / Heidelberg.

Blei, D. M. 2012. Probabilistic topic models. *Communications of the ACM* 55 (4): 77–84.

Blei, D. M. and Lafferty, J. D. 2007. A correlated topic model of science. *The Annals of Applied Statistics* 1 (1): 17–35.

Blei, D. M. and Lafferty, J. D. 2009a. Topic Models. In *Text mining: classification, clustering, and applications,* ed. A. Srivastava and M. Sahami. Data Mining and Knowledge Discovery Series. Boca Raton, Florida, USA: Chapman / Hall/CRC.

———. 2009 b. Visualizing Topics with Multi-Word Expressions. *ArXiv e-prints,* no. arXiv:0907.1013v1.

Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3:993–1022.

Blei, D. M., Griffiths, T. L., Jordan, M. I., and Tenenbaum, J. B. 2004. Hierarchical Topic Models and the Nested Chinese Restaurant Process. In *Advances in Neural Information Processing Systems 16,* ed. S. B. Thrun S. Saul L.K. Cambridge, Massachusetts, USA: The MIT Press.

Bordag, S. 2007. Elements of Knowledge-free and Unsupervised Lexical Acquisition. PhD diss., Universität Leipzig.

Boyd-Graber, J., Blei, D. M., and Zhu, X. 2007. A Topic Model for Word Sense Disambiguation. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL),* 1024–1033. Prague, Czech Republic.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. 2002. TIGER Treebank. In *Proceedings of the workshop on treebanks and linguistic theories (TLT02).* Sozopol, Bulgaria.

Bronstein, I. N., Semendjajew, K. A., Musiol, G., and Mühlig, H. 2005. *Taschenbuch der Mathematik.* 6th ed. Frankfurt a.M., Germany: Verlag Harri Deutsch.

Burchardt, A., Erk, K., Frank, A., Kowalski, A., Padó, S., and Pinkal, M. 2006. The SALSA Corpus: a German corpus resource for lexical semantics. In *Proceedings of the 5th international conference on language resources and evaluation (LREC-2006).* Genoa, Italy.

Cai, J., Lee, W. S., and Teh, Y. W. 2007. Improving Word Sense Disambiguation Using Topic Features. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL),* 1015–1023. Prague, Czech Republic.

Carthy, J. 2004. Lexical Chains versus Keywords for Topic Tracking. In *Computational linguistics and intelligent text processing,* ed. A. Gelbukh, 507–510. Vol. 2945. Lecture Notes in Computer Science. Berlin / Heidelberg: Springer.

Cover, T. M. and Thomas, J. A. 1991. *Elements of information theory.* Hoboken, New Jersey, USA: Wiley.

Cramer, I., Finthammer, M., Kurek, A., Sowa, L., Wachtling, M., and Claas, T. 2008. Experiments on Lexical Chaining for German Corpora: Annotation, Extraction, and Application. *Journal for Language Technology and Computational Linguistics (JLCL)* 23 (2): 34–48.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41 (6): 391–407.

Dinu, G. and Lapata, M. 2010. Topic Models for Meaning Similarity in Context. In *COLING '10: Proceedings of the 23rd International Conference on Computational Linguistics: Posters,* 250–258. Beijing, China.

Dunning, T. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics* 19 (1): 61–74.

Ercan, G. and Cicekli, I. 2007. Using lexical chains for keyword extraction. *Inf. Process. Manage.* 43 (6): 1705–1714.

Evert, S. 2005. The Statistics of Word Cooccurrences: Word Pairs and Collocations. PhD diss., Universität Stuttgart.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database.* Language, Speech, and Communication. Cambridge, Massachusetts, USA: The MIT Press.

Gale, W. A., Church, K. W., and Yarowsky, D. 1992. One Sense Per Discourse. In *Proceedings of the workshop on speech and natural language,* 233–237. Harriman, New York, USA.

Galley, M. and McKeown, K. 2003. Improving word sense disambiguation in lexical chaining. In *IJCAI'03: proceedings of the 18th international joint conference on artificial intelligence,* 1486–1488. Acapulco, Mexico.

Girolami, M. and Kabán, A. 2003. On an Equivalence between PLSI and LDA. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 433–434. Toronto, Canada.

Gong, Y. and Liu, X. 2001. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 19–25. New Orleans, Louisiana, USA.

Green, S. J. 1996. Using Lexical Chains to Build Hypertext Links in Newspaper Articles. In *AAAI-96 workshop on internet-based information systems,* 115–141. Portland, Oregon, USA.

Griffiths, T. L. and Steyvers, M. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America* 101 (Suppl 1): 5228–5235.

Al-Halimi, R. and Kazman, R. 1998. Temporal Indexing through Lexical Chaining. In *WordNet: an electronic lexical database,* ed. C. Fellbaum, 333–351. Language, Speech, and Communication. Cambridge, Massachusetts, USA: The MIT Press.

Halliday, M. A. K. and Hasan, R. 1976. *Cohesion in English.* English language series. London: Longman.

Hamp, B. and Feldweg, H. 1997. GermaNet - a Lexical-Semantic Net for German. In *Proceedings of the ACL/EACL-97 workshop automatic information extraction and building of lexical semantic resources for NLP applications.* Madrid, Spain.

Hasan, R. 1984. Coherence and Cohesive Harmony. In *Understanding Reading Comprehension,* ed. J. Flood, 181–220. Cognition, Language, and the Structure of Prose. Newark, Delaware, USA: International Reading Association.

Hearst, M. A. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd annual meeting on association for computational linguistics,* 9–16. ACL '94. Las Cruces, New Mexico, USA.

Heinrich, G. 2009. *Parameter estimation for text analysis v2.9.* Technical report. Darmstadt, Germany: Fraunhofer IGD.

Hennig, L. 2009. Topic-based multi-document summarization with probabilistic latent semantic analysis. In *Proceedings of the international conference RANLP-2009,* 144–149. Borovets, Bulgaria.

Hirst, G. and St-Onge, D. 1998. Lexical Chains as representation of context for the detection and correction malapropisms. In *WordNet: An Electronic Lexical Database,* ed. C. Fellbaum, 305–332. Language, Speech, and Communication. Cambridge, Massachusetts, USA: The MIT Press.

Hoey, M. 1991. *Patterns of Lexis in Text.* Describing English Language. Oxford, UK: Oxford University Press.

Hoffman, M. D., Blei, D. M., and Bach, F. R. 2010. Online Learning for Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 23,* ed. J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, 856–864. Cambridge, Massachusetts, USA: The MIT Press.

Hoffman, M., Blei, D. M., Wang, C., and Paisley, J. 2012. Stochastic Variational Inference. *ArXiv e-prints,* no. arXiv:1206.7051v1.

Hofmann, T. 1999a. Probabilistic Latent Semantic Analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence,* 289–296. UAI '99. Stockholm, Sweden.

———. 1999 b. Probabilistic Latent Semantic Indexing. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 50–57. Berkeley, California, USA.

Hollingsworth, W. and Teufel, S. 2005. Human annotation of lexical chains: Coverage and agreement measures. In *Proceedings of the Workshop ELECTRA: Methodologies and Evaluation of Lexical Cohesion Techniques in Real-world Applications.* In Association with SIGIR '05. Salvador, Brazil.

Hubert, L. and Arabie, P. 1985. Comparing partitions. *Journal of Classification* 2 (1): 193–218.

Jakob, N. 2007. Developing a generic interface for semantic networks. Diploma thesis, Technische Universität Darmstadt.

Jarmasz, M. 2003. Roget's thesaurus as a lexical resource for natural language processing. Master's thesis, University of Ottawa.

Lin, C.-Y. and Hovy, E. 2000. The Automated Acquisition of Topic Signatures for Text Summarization. In *COLING '00: Proceedings of the 18th Conference on Computational Linguistics,* 495–501. Vol. 1. Saarbrücken, Germany.

Manning, C., Raghavan, P., and Schütze, H. 2008. *An Introduction to Information Retrieval.* Cambridge, UK: Cambridge University Press.

Manning, C. and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing.* Cambridge, Massachusetts, USA: The MIT Press.

Marathe, M. and Hirst, G. 2010. Lexical Chains Using Distributional Measures of Concept D. In *Proceedings of the 11th International Conference on Computational Linguistics and Intelligent Text Processing,* 291–302. CICLing'10. Iaşi, Romania.

Medelyan, O. 2007. Computing lexical chains with graph clustering. In *Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop,* 85–90. ACL '07. Prague, Czech Republic.

Meilă, M. 2003. Comparing clusterings by the variation of information. In *Proceedings of the 16th Annual Conference of Computational Learning Theory (COLT).* Washington, D.C., USA.

———. 2005. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd International Conference on Machine Learning,* 577–584. ICML '05. Bonn, Germany.

———. 2007. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis* 98 (5): 873–895.

Menestrina, D., Whang, S. E., and Garcia-Molina, H. 2010. Evaluating entity resolution results. *Proceedings of the VLDB Endowment* 3 (1): 208–219.

Milgram, S. 1967. The Small World Problem. *Psychology Today* 67 (1): 61–67.

Misra, H., Yvon, F., Jose, J., and Cappé, O. 2009. Text Segmentation via Topic Modeling: An Analytical Study. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management,* 1553–1556. CIKM 2009. Hong Kong, China.

Moldovan, D. and Novischi, A. 2002. Lexical chains for question answering. In *COLING '02: Proceedings of the 19th International Conference on Computational Linguistics,* 1–7. Taipei, Taiwan.

Morris, J. 2010. Individual Differences in the Interpretation of Text: Implications for Information Science. *Journal of the American Society for Information Science and Technology (JASIST)* 61 (1): 141–149.

Morris, J. and Hirst, G. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 17:21–48.

———. 2004. The Subjectivity of Lexical Cohesion in Text. In *Proceedings of the AAAI spring symposium on exploring attitude and affect in text: theories and applications.* Palo Alto, California, USA.

Müller, C. and Strube, M. 2006. Multi-level annotation of linguistic data with MMAX2. In *Corpus technology and language pedagogy: new resources, new tools, new methods,* ed. S. Braun, K. Kohn, and J. Mukherjee, 197–214. Frankfurt a.M., Germany: Peter Lang.

Nelken, R. and Shieber, S. M. 2007. *Lexical Chaining and Word-Sense-Disambiguation.* TR-06-07. Technical Report. Cambridge, Massachusetts, USA: Harvard University.

Newman, D., Asuncion, A., Smyth, P., and Welling, M. 2009. Distributed Algorithms for Topic Models. *Journal of Machine Learning Research* 10:1801–1828.

Okumura, M. and Honda, T. 1994. Word sense disambiguation and text segmentation based on lexical cohesion. In *COLING '94: Proceedings of the 15th Conference on Computational Linguistics,* 755–761. Vol. 2. Kyoto, Japan.

Phan, X.-H. and Nguyen, C.-T. 2007. GibbsLDA++: A C/C++ implementation of latent Dirichlet allocation (LDA). http://gibbslda.sourceforge.net.

Quasthoff, U., Richter, M., and Biemann, C. 2006. Corpus Portal for Search in Monolingual Corpora. In *Proceedings of the 5th international conference on language resources and evaluation (LREC-2006).* Genoa, Italy.

Rand, W. M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 66 (336): 846–850.

Reeve, L., Han, H., and Brooks, A. D. 2006. Biochain: Lexical Chaining Methods for Biomedical Text Summarization. In *Proceedings of the 2006 ACM symposium on applied computing (SAC 2006),* 180–184. Dijon, France.

Reichart, R. and Rappoport, A. 2009. The NVI clustering evaluation measure. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning,* 165–173. CoNLL '09. Boulder, Colorado, USA.

Riedl, M. and Biemann, C. 2012. Sweeping through the Topic Space: Bad luck? Roll again! In *ROBUS-UNSUP 2012: Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP held in conjunction with EACL 2012,* 19–27. Avignon, France.

Roget, P. M. 1852. *Roget's thesaurus of english words and phrases.* Harlow, UK: Longman Group Ltd.

Rosenberg, A. and Hirschberg, J. 2007. V-measure: a conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in Natural Language Processing and computational natural language learning (EMNLP-CoNLL),* 410–420. Prague, Czech Republic.

Schwager, F. 2008. Automatic analysis of lexical cohesion. Diploma thesis, Technische Universität Darmstadt.

Silber, H. G. and McCoy, K. F. 2002. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics* 28 (4): 487–496.

Stairmand, M. A. 1996. A computational analysis of lexical cohesion with applications in information retrieval. PhD diss., Center for Computational Linguistics, UMIST.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. 2012. Brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the demonstrations session at EACL 2012.* Avignon, France.

Steyvers, M. and Griffiths, T. L. 2006. Probabilistic topic models. In *Latent semantic analysis: a road to meaning.* Ed. T. Landauer, D. Mcnamara, S. Dennis, and W. Kintsch. Laurence Erlbaum.

Stokes, N., Carthy, J., and Smeaton, A. F. 2004. SeLeCT: A Lexical Cohesion Based News Story Segmentation System. *AI Communications* 17 (1): 3–12.

Strehl, A. 2002. Relationship-based clustering and cluster ensembles for high-dimensional data mining. PhD diss., University of Texas.

Strehl, A. and Ghosh, J. 2003. Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research* 3:583–617.

Stührenberg, M., Goecke, D., Diewald, N., Mehler, A., and Cramer, I. 2007. Web-based annotation of anaphoric relations and lexical chains. In *Proceedings of the Linguistic Annotation Workshop (the LAW),* 140–147. Prague, Czech Republic.

Teh, Y. W., Newman, D., and Welling, M. 2007. A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 19,* ed. B. Schölkopf, J. Platt, and T. Hoffman, 1353–1360. Cambridge, Massachusetts, USA: The MIT Press.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. 2006. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association* 101 (476): 1566–1581.

Teich, E. and Fankhauser, P. 2005. Exploring Lexical Patterns in Text: Lexical Cohesion Analysis with WordNet. In *Heterogeneity in Focus: Creating and Using Linguistic Databases,* 129–145. Vol. 2. Interdisciplinary Studies on Information Structure (ISIS). Potsdam: Universitätsverlag Potsdam.

Tomokiyo, T. and Hurst, M. 2003. A language Model Approach to Keyphrase Extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment,* 33–40. Vol. 18. MWE ’03. Sapporo, Japan.

Wallach, H. M. 2006. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning,* 977–984. ICML ’06. Pittsburgh, Pennsylvania, USA.

Wang, X. and McCallum, A. 2006. Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 424–433. KDD 2006. Philadelphia, Pennsylvania, USA.

Wang, X., McCallum, A., and Wei, X. 2007. Topical n-grams: phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining,* 697–702. ICDM ’07. Omaha, Nebraska, USA.

Yang, D. and Powers, D. M. W. 2006. Word Sense Disambiguation using lexical cohesion in the context. In *Proceedings of the COLING/ACL 2006 main conference poster sessions,* 929–936. Sydney, Australia.

A Proof: Equality of NMI and V

to show:

$$V(C,K) \equiv NMI(C,K) \tag{A.1}$$

with

$$V(C,K) = 2 \times \frac{h \times c}{h + c} \tag{A.2}$$

$$h = 1 - \frac{H(C|K)}{H(C)} \quad , \quad c = 1 - \frac{H(K|C)}{H(K)} \tag{A.3}$$

and

$$NMI(C,K) = 2 \times \frac{I(C,K)}{H(C) + H(K)} \tag{A.4}$$

reformulate h and c using the fact that $I(C,K) = H(C) - H(C|K) = H(K) - H(K|C)$:

$$\begin{aligned} h &= 1 - \frac{H(C|K)}{H(C)} \\ &= \frac{H(C)}{H(C)} - \frac{H(C|K)}{H(C)} \\ &= \frac{H(C) - H(C|K)}{H(C)} \\ &= \frac{I(C,K)}{H(C)} \end{aligned} \tag{A.5}$$

$$\begin{aligned} c &= 1 - \frac{H(K|C)}{H(K)} \\ &= \frac{H(K)}{H(K)} - \frac{H(K|C)}{H(K)} \\ &= \frac{H(K) - H(K|C)}{H(K)} \\ &= \frac{I(C,K)}{H(K)} \end{aligned} \tag{A.6}$$

simplifying $h \times c$:

$$\begin{aligned} h \times c &= \frac{I(C,K)}{H(C)} \times \frac{I(C,K)}{H(K)} \\ &= \frac{I(C,K)^2}{H(C)H(K)} \end{aligned} \tag{A.7}$$

simplifying $h + c$:

$$\begin{aligned} h + c &= \frac{I(C,K)}{H(C)} + \frac{I(C,K)}{H(K)} \\ &= \frac{I(C,K)H(K)}{H(C)H(K)} + \frac{I(C,K)H(C)}{H(C)H(K)} \\ &= \frac{I(C,K)(H(K) + I(C,K)H(C)}{H(C)H(K)} \\ &= \frac{I(C,K)[(H(K) + H(C)]}{H(C)H(K)} \end{aligned} \tag{A.8}$$

simplifying $\frac{h \times c}{h+c}$ using (A.7) and (A.8):

$$\begin{aligned} \frac{h \times c}{h + c} &= \frac{\dfrac{I(C,K)^2}{H(C)H(K)}}{\dfrac{I(C,K)[(H(K) + H(C)]}{H(C)H(K)}} \\ &= \frac{I(C,K)^2}{H(C)H(K)} \times \left[\frac{I(C,K)[H(K) + H(C)]}{H(C)H(K)} \right]^{-1} \\ &= \frac{I(C,K)^2}{H(C)H(K)} \times \frac{H(C)H(K)}{I(C,K)[(H(K) + H(C)]} \\ &= \frac{I(C,K)}{H(K) + H(C)} \end{aligned} \tag{A.9}$$

substituting (A.9) into (A.2) shows that NMI and V are equal:

$$\begin{aligned} V(C,K) &= 2 \times \frac{h \times c}{h + c} \\ &= 2 \times \frac{I(C,K)}{H(K) + H(C)} \\ &= NMI(C,K) \end{aligned} \tag{A.10}$$

□

B Annotation Guidelines

— Lexical Chain Annotation Guidelines —

Steffen Remus

v1.0 – September 1, 2012

About the document

Use this document as a schema to identify and annotate lexical chains in documents of the current *Salsa*[1] *2.0 / TiGer*[2] *2.1 Corpus*. This document guides you through common standards and potential pitfalls.

Introduction to lexical chains

When reading a text, whether short or long, the text is intelligible because of its structure. Each text is designed to transport a message, and a reasonable structure is the essential means for this. *Lexical chains* visualize this structure by connecting words or phrases in the text that are semantically related. These words or phrases are called *lexical items* and each item contributes to a specific meaning of a lexical chain. Consider the given example text below and some extracted lexical chains:

> COFFEE QUOTA TALKS CONTINUE BUT NO AGREEMENT YET
>
> LONDON, March 2 - Coffee quota talks at the International Coffee Organization council meeting here continued this afternoon, but producers and consumers still had not reached common ground on the key issue of how to estimate export quotas, delegates said.
>
> The 54 member contact group was examining a Colombian proposal to resume quotas April 1 under the ad hoc system used historically, with a pledge to meet again in September to discuss how quotas would be worked out in the future, they said.

[1] SALSA – The Saarbrücken Lexical Semantics Acquisition Project, "see A. Burchardt et al. 2006. The SALSA Corpus: a German corpus resource for lexical semantics. In *Proceedings of the 5th international conference on language resources and evaluation (LREC-2006).* Genoa, Italy".

[2] TiGer – TiGer Treebank (Version 2.1), see "S. Brants et al. 2002. TIGER Treebank. In *Proceedings of the workshop on treebanks and linguistic theories (TLT02).* Sozopol, Bulgaria".

1. { *COFFEE QUOTA TALKS, Coffee quota talks, Coffee Organization council meeting, quotas, quotas, quotas* }
2. { *CONTINUE, continued, resume* }
3. { *AGREEMENT, reached common ground, pledge, discuss* }
4. { *producers, consumers* }

Related terms are underlined with the same color, which build up the chains 1-4.

Although the example is incomplete yet (meaning there are more chains in the text), the central theme of the text (by just looking at the chains) clearly can be identified.

This technique can be utilized in many applications, and the main goal is to automatically identify these chains with the help of computer systems. In order to evaluate these systems, i.e. to say how good or bad they did the job, we need examples annotated by humans and compare them with those made by the system.

Short Description

- Lexical items are the basic elements of the chains occurring in the texts.
- Candidates for lexical items are words or phrases bearing some meaning.
- Lexical items describing the same concept or relating to the same topic form a so-called dense chain.
- Within dense chains each item must be of the same topic as every other item in the chain.
- Each lexical item must solely exist in exactly one chain.
- Chains consisting of only a single item may exist.
- Each dense chain can be assigned a description of its concept.
- Dense chains may be linked to describe broader concepts.
- Source and target of links are representative lexical items, which may link to several dense chains.

Schema

A Candidate items (items to annotate)

The question of which words or phrases in the text are adequate lexical items is crucial for forming lexical chains.

In general we will only consider **terms or phrases associated with meaning** as possible candidates, specifically:

- **Nouns** and **compounds** (examples for compounds are *cellar door*, *city council member*, *small talk*, ...).
- **Adjective noun combinations**, where the modifying adjective significantly influences the meaning of the item. E.g., the topic of *money* may be *economics*, but the phrase *dirty money* is more in the domain of *cheating* or more general *crime*.

  Always **prefer shorter expressions to longer ones**. Only consider longer phrases when the concept of the shorter phrase is very different and not cohesive at all. For example in the sentence `'Dorothy ate a hearty supper ...'` the phrase *hearty supper* is of minor interest, but the *supper* itself maybe indeed important for the context and an adequate lexical item.
- **Ignore words that bear no relevant meaning**. Strictly speaking, words that bring no relevant information about the context. These are typically words that frequently occur in texts. Examples for this include: *a*) numbers (*once, twice, hundreds, million, ...*) *b*) units (*percent, miles, meters, gallons, ...*) *c*) currencies (*dollar, euro, yen, ...*) *d*) times / periods (*monday, year, quarter, ...*) *e*) frequently occurring words (so called stop words) (*everybody*, *anybody*, *person*, *name*, *good*, *bad*, *start*, *end*, *go*, *walk*, ...) Specifically **ignore names, places and referring items** (*John, he, Toyota, it, ...*).

B Level one: dense lexical chains

- **Each item in a specific chain must be of the same topic as every other item in the chain.** This implies: *each chain can be assigned exactly one topical description.* More precisely: item X is of the same *narrow* topic as every other item Y in the chain. E.g. *car* is related to *driver* because they share the same topic (for instance *traffic* or *transport*) and every other item in the chain, e.g. the term *leg*, must also share this specific topic, as would obviously not be the case here.

Definition: *a*) **collocational** terms (i.e. semantic relationships of words / items that often co-occur) *b*) **repetition** of words or phrases (also including words with different flection (e.g. *tourist*, *tourists*)) *c*) **synonyms** or **near-synonyms** *d*) **superordinates** or **subordinates** (cf. hypernyms and hyponyms, e.g. *mammal* is a hypernym / superordinate term of *cow*) *e*) **opposites** or **antonyms** (also including converseness (doctor – patient), incompatibility, complementarity (male – female), ...) *f*) **part-whole-relations** (cf. meronymy, e.g. *arm*, *leg* are meronyms of *body*)

- Unfortunately the previous definition is accompanied with transitivity and uncertainty. As an example, consider the terms *car*, *bus*, and *airplane*. All of them share the same topic *vehicles*, thus one could create one dense chain {*car, bus, airplane*}. But *car* and *bus* are also *driving vehicles* whereas *airplane* is a *flying vehicle*, so one could also create two chains {*car*, *bus*} and {*airplane*}. On the other hand, unlike *car*, *bus* and *airplane* are both means of *public transport*, which would then result in {*bus, airplane*} and {*car*}.

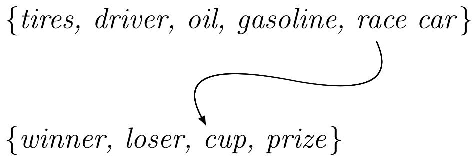
  This example shows that the manual annotation process can only be done by human **intuition** (i.e. common sense and knowledge of the language) The context of candidate items is always of enormous importance and determinant for the choice of the chain. Always consider the following rule of thumb:

  > *If you "feel" that two candidate items are "semantically too far away" then better prefer a level two link to the broader item.*

C Level two: links between dense chains

- **Dense chains can be linked via lexical items.** A link can be described as a semantic relation between two lexical items belonging to different dense chains. As in B, a semantic relationship exist if two lexical items share the same topic.

  This concept is best described with an example, consider therefore the term *race car* where the head of the compound signals the topic *car* (in general) and the modifier signals the topic *racing* or *contest*. Consider further the two fictional chains {*tires, driver, oil, gasoline*} describing the topic *car* and {*winner, loser, cup, prize*} describing the topic *contest*. The term *race car* belongs in principle into both chains, but its stronger affinity belongs to the former one (*car*), and since in our scenario a lexical item belongs to exactly one chain it will be put into the *car* chain. Stopping here would lead to inconsistencies, so we connect *race car* with the *contest* chain via that lexical item, that is most coherent with *race car* (i.e. the most representative lexical item sharing the same topic), here *cup*, visually resulting in:



{*tires, driver, oil, gasoline, race car*}

{*winner, loser, cup, prize*}

- Once two chains are connected there is no need to add further links. Each chain corresponds to its own description of a concept, and the links between chains express that these concepts are related and form a wider concept. E.g. the resulting description of the concept of the linked chains in the example above would be *auto racing*.
- Several links via the same lexical item are possible and allowed. E.g. think about *formula one race car* which may encode three topics.

D Potential pitfalls

- homonymy or polysemy – i.e. same form different meaning / topic.
- items share the same topic in general, but not in the current context. Consider for example the following two extracts of the same document:

  "... due to the falsification of documents, ..."

  and

  "... quoted the state department spokesman from the letter."

  Although the word *documents* is in fact a hypernym of the word *letter*, they should not be considered to be put into the same dense chain, because in this special case the context of the term *letter* refers to some topic like *writing*, whereas *documents* refers to something *official* e.g. ID cards, passports, or the like.
- Beware of the correct meaning of compounds, e.g. which part is the head and which the modifier, or is the meaning of the compound completely independent. Consider for example the words *waterbed* and *blackboard*, where *waterbed* is simply a special kind of *bed* but *blackboard* is not simply a *black board*.

Annotation recipe

READ – GRASP – SEARCH & MERGE – REVISIT & LINK

1. **Read** the document and **grasp** the main message behind the text.
2. **Pick** the next unchained lexical item as described in A.
3. **Search** for appropriate chains or other lexical items, and if they share the same topic as described in B **merge** them.

   Remember that **each item** in the dense chain should satisfy the condition **with every other** item in the chain.
4. Continue with 2 until all lexical items in the text are served.

   Remember that single item chains may exist.
5. **Revisit** the dense chains and check if you should do one of the following operations:
   - splitting of chains – often chains grow too large because a too broad topic was chosen.
   - merging of chains – often too many chains emerge because a too narrow topic was chosen.
6. **Add links** between dense chains via the lexical items as described in C. Try to **pick meaningful representatives**, because it often happens that many items serve as candidate links. It is thus sufficient to **link chains only once**. In other words, if a link exists between the chains A and B there's no need to seek for other links between A and B.

   Remember that several links via the same lexical item are allowed.

When you are done, you ideally have one or two "big chains" (when considering linked chains as a whole) describing the main message of the text, as well as a few small chains describing side topics.

Examples

The following examples show selected texts from the *Salsa* corpus. Terms in square brackets are preselected candidate lexical items (cf. schema section A). This preselection is done to ease the annotation process, schema A in particular. These items are automatically selected based on a part-of-speech pattern and are not necessarily exact. However, in most cases the candidate items are adequate enough and no effort must be put into finding additional candidate items. Consider only correct items for chaining and linking, though.

The examples comprise the original text with preselected candidate items, the identified dense chains and links including comments. The examples are supplemental to the annotation recipe above and may still be arguable, though they might give a good direction to reach common ground.

TEXT:

USA

Weiteres [Geständnis] im [Touristenmord-Prozeß]

RIVERSIDE , 17. November (ap / dpa) .

Der dritte [Beteiligte] an der [Ermordung] der [deutschen [Touristin]] Gisela Pfleger hat sich des [Raubes] und [Totschlags] schuldig bekannt .

Die [Staatsanwaltschaft] in Riverside in Kalifornien machte das [Bekenntnis] des [20jährigen] Xou Yang jetzt publik .

Des [Mordes] an der [64jährigen] aus Emmerich sind bereits der 19jährige Thongxay Nilakout und der 20jährige Khamchan Bret Ketsouvannasane schuldig gesprochen worden .

Die [Strafmaße] werden später bekanntgegeben .

Zu [lebenslangen [Freiheitsstrafen]] wurden zwei [junge [US-Amerikaner]] in Miami wegen des [Mordes] an dem [deutschen [Touristen]] Uwe-Wilhelm Rakebrand verurteilt .

DENSE CHAINS:

1. {[Geständnis], [Bekenntnis]}
   (the usage of “Geständnis” and “Bekenntnis” is synonymous)
2. {[Touristenmord-Prozeß], [Staatsanwaltschaft], [Strafmaße], [Freiheitsstrafen]}
   (All items share the same topic (e.g. “Prozeß”). “Touristenmord-Prozeß” is an endocentric compound consisting of “Tourist”, “Mord”, and “Prozeß”, but the predominant part is “Prozeß”. The fact that in this case “Freiheitsstrafen” is favored over “lebenslangen Freiheitsstrafen” is that the adjective “lebenslang” is not relevant to get the message of the text.)
3. {[Ermordung], [Raubes], [Totschlags], [Mordes], [Mordes]}
   (“Ermordung” and “Mordes” can be used synonymous and all words share the same topic (e.g. “Verbrechen”))
4. {[Touristin], [Touristen]}
   (Same words but different flection. “Touristin” and “Tourist” are favored over “deutschen Touristin” and “deutschen Touristen” because the modifying adjective is irrelevant.)

LINKS:

1. {Touristenmord-Prozeß → Geständnis }
   (A trial (“Prozeß”) often ends with a confession (“Geständnis”), thus the terms are related.)
2. {Touristenmord-Prozeß → Ermordung}
   (“Ermordung” is synonymous with “Mord” which is a part of the compound “Touristenmord-Prozeß”)
3. {Touristenmord-Prozeß → Touristen}
   (“Touristen” is a part of the compound “Touristenmord-Prozeß’.’)
4. {Ermordung → Beteiligte}
   (“Ermordung” is a criminal act with at least two participants (“Beteiligte”) the murderer and the victim, in this case two murderers.)

TEXT:

[Franzosen] gegen [Atomtests]

PARIS (ap) .

Zwei von drei [Franzosen] sind nach einer neuen [Meinungsumfrage] gegen die [Atomversuche] ihres [Landes] .

Nach den am Montag in Paris [veröffentlichten [Ergebnissen]] des [Instituts] TMO Consultants sprachen sich 23 Prozent für die [Versuche] im Südpazifik aus - zehn Prozent äußerten keine [Meinung] .

DENSE CHAINS:

1. {[Franzosen],[Franzosen]}
   (Repetion of the same word.)
2. {[Atomtests],[Atomversuche],[Versuche]}
   (Each word is about a test (nuclear tests in particular).)
3. {[Meinungsumfrage],[Ergebnissen],[Meinung]}
   (Each word is about the same topic which is some opinion poll.)

LINKS:

1. {[Franzosen] → [Landes]}
   (The term "Landes" builds up a single element chain and is linked to "Franzosen" because it is France which is meant by the term "Landes".)
2. {[Meinungsumfrage] → [Instituts]}
   ("Instituts" builds up a single chain element and is linked to "Meinungsumfrage" because a poll is mostly carried out by institutions.)

TEXT:

[Lärm] gegen (Ver) [schweigen]

BELGRAD , 3. Januar (rtr) .

Tausende [Bewohner] der [jugoslawischen [Hauptstadt]] Belgrad sind dem [Aufruf] der [Opposition] gefolgt und haben am Donnerstag während der [TV-Abendnachrichten] mit [Töpfen] und [[Pfannen] [Lärm]] gemacht .

[Studenten] hatten zu dem [Protest] aufgerufen , um gegen die einseitige [Berichterstattung] des [staatlichen [Fernsehens]] zu protestieren .

Der [Krach] war in ganz Belgrad zu hören .

Die [Opposition] warf der [Stadtverwaltung] vor , das [Streuen] [vereister [Straßen]] absichtlich zu verzögern .

Dadurch kam es zu vielen [Knochenbrüchen] .

Erstmals hat auch die [Serbisch-Orthodoxe [Kirche]] der [[Regierung] [Wahlbetrug]] vorgeworfen .

DENSE CHAINS:

1. {[Lärm],[schweigen],[Lärm],[Krach]}
   (The main shared topic of these terms is noise. Here, the term "schweigen" is only partly correct extracted (cf. "(Ver) schweigen"), but the message is the same.)
2. {[Bewohner],[Hauptstadt]}
   (Each town has citizens.)
3. {[Aufruf],[Protest]}
   (The meaning of the term "Aufruf" is in fact "Protestaufruf", which is in the same topic as "Protest".)
4. {[Opposition],[Opposition],[Stadtverwaltung],[Regierung]}
   (An example were each term is connected with the topic "politics".)
5. {[TV-Abendnachrichten],[Berichterstattung],[Fernsehens]}
   (Each term is about public media.)
6. {[Streuen],[vereister Straßen]}
   (Icy streets should be salted.)

LINKS:

1. {Protest → Lärm}
   (Most protests are noisy.)
2. {Studenten → Protest}
   (In most protests students are involved.)
3. {vereister Straßen → Knochenbrüchen}
   (Not salting icy streets increases the probability of injuries of any kind.)
4. {Regierung → Wahlbetrug}
   (This link tells that the government is playing wrong indicated by the last part of the compound "Wahlbetrug", where the first part of the compound ("Wahl") is directly connect to the chain describing the topic politics.)

C Annotation Guidelines Companion

— Lexical Chain Annotation Guidelines Companion —

to be accompanied with “Lexical Chain Annotation Guidelines v1.0”

Steffen Remus

v1.0 – September 1, 2012

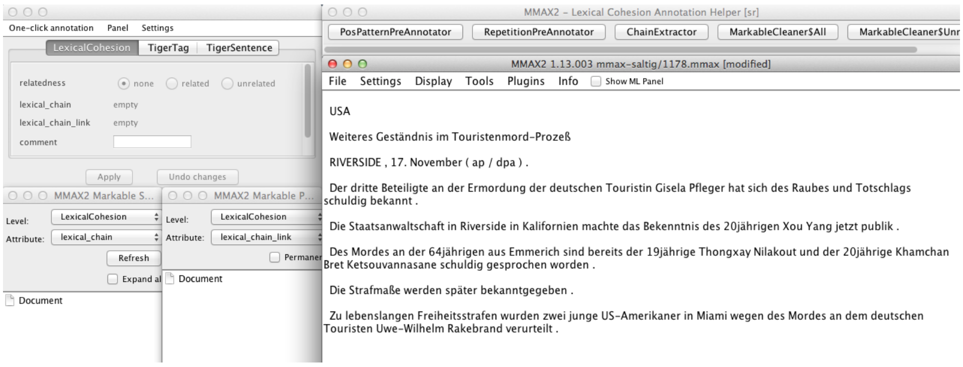
Annotating Lexical Chains with MMAX2[1]

Prepare the annotation project by extracting the file “annotate-distr.zip” into any directory of your choice, open the file “Annotation.properties”, search for the line “annotator-id=$USER” and replace $USER by your id (e.g. your initials), save and close the file. Open MMAX2 via command line (preferred):

- on MS Windows open the command prompt , navigate to the extracted directory, and enter the command “open.bat mmax-saltig/XXXX.mmax”, where XXXX is the document id of the document you want to annotate.
- on Linux, Unix, MacOSX open the terminal, navigate to the extracted directory, and enter the command “./open.sh mmax-saltig/XXXX.mmax”, where XXXX is the document id of the document you want to annotate.
- alternatively you may want to run the program by clicking MMAX2Interface.jar (thus running it as Java application), then open the desired document by clicking “File->load” and navigate to the desired .mmax document.

The following screenshots are based on document 1178. After opening it the interface should look like this:

[1] http://mmax2.sourceforge.net or http://mmax2.net — C. Müller and M. Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In *Corpus technology and language pedagogy: new resources, new tools, new methods,* ed. S. Braun et al., 197–214. Frankfurt a.M., Germany: Peter Lang



relatedness ● none ○ related ○ unrelated
lexical_chain empty
lexical_chain_link empty
comment

Level: LexicalCohesion
Attribute: lexical_chain

Level: LexicalCohesion
Attribute: lexical_chain_link

USA

Weiteres Geständnis im Touristenmord-Prozeß

RIVERSIDE , 17. November (ap / dpa) .

Der dritte Beteiligte an der Ermordung der deutschen Touristin Gisela Pfleger hat sich des Raubes und Totschlags schuldig bekannt .

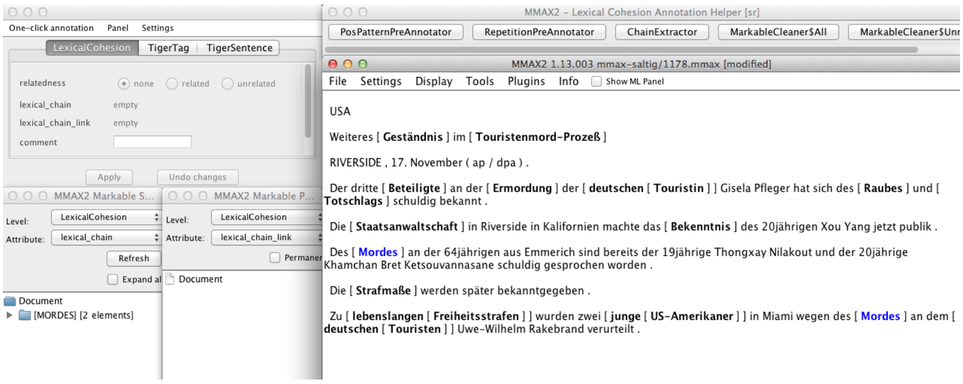
Die Staatsanwaltschaft in Riverside in Kalifornien machte das Bekenntnis des 20jährigen Xou Yang jetzt publik .

Des Mordes an der 64jährigen aus Emmerich sind bereits der 19jährige Thongxay Nilakout und der 20jährige Khamchan Bret Ketsouvannasane schuldig gesprochen worden .

Die Strafmaße werden später bekanntgegeben .

Zu lebenslangen Freiheitsstrafen wurden zwei junge US-Amerikaner in Miami wegen des Mordes an dem deutschen Touristen Uwe-Wilhelm Rakebrand verurteilt .

You can now proceed with the annotation process as described in the annotation recipe in the annotation guidelines. When done READING & GRASPING click on "PosPatternPreAnnotator" and "RepetitionPreAnnotator" in this order. The preselected candidate items (called *markables* henceforth) will instantly appear in the text, and repeated markables are already chained together. See here:



relatedness ● none ○ related ○ unrelated
lexical_chain empty
lexical_chain_link empty
comment

Level: LexicalCohesion
Attribute: lexical_chain

Level: LexicalCohesion
Attribute: lexical_chain_link

USA

Weiteres [**Geständnis**] im [**Touristenmord-Prozeß**]

RIVERSIDE , 17. November (ap / dpa) .

Der dritte [**Beteiligte**] an der [**Ermordung**] der [**deutschen** [**Touristin**]] Gisela Pfleger hat sich des [**Raubes**] und [**Totschlags**] schuldig bekannt .

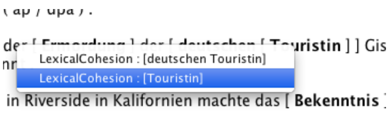
Die [**Staatsanwaltschaft**] in Riverside in Kalifornien machte das [**Bekenntnis**] des 20jährigen Xou Yang jetzt publik .

Des [**Mordes**] an der 64jährigen aus Emmerich sind bereits der 19jährige Thongxay Nilakout und der 20jährige Khamchan Bret Ketsouvannasane schuldig gesprochen worden .

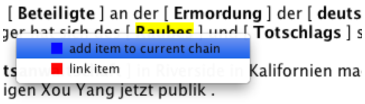
Die [**Strafmaße**] werden später bekanntgegeben .

Zu [**lebenslangen** [**Freiheitsstrafen**]] wurden zwei [**junge** [**US-Amerikaner**]] in Miami wegen des [**Mordes**] an dem [**deutschen** [**Touristen**]] Uwe-Wilhelm Rakebrand verurteilt .

Select a markable by clicking on one of the square brackets surrounding it, or click directly on the word. When the word is spanned by more than one markable a menu will appear where you have to select the desired markable. E.g. in the example below the word "Touristin" is spanned by two markables ("deutsche Touristin" and "Touristin").

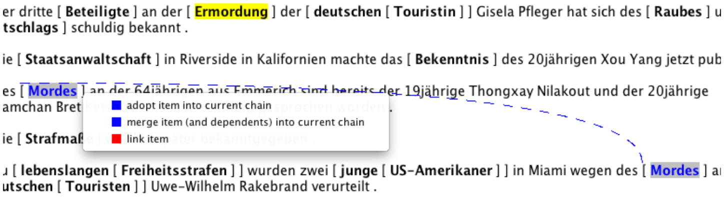

(ap / dpa) .
der [Ermordung] der [deutschen [Touristin]] Gis
LexicalCohesion : [deutschen Touristin]
LexicalCohesion : [Touristin]
in Riverside in Kalifornien machte das [Bekenntnis

In order to chain or link markables, select first a markable (no matter if it is in a chain or not) and it will instantly appear shaded yellow. Then right click on another markable that you want to chain or link with, and select the desired action (adding to chain, or linking). When right clicking on the destination markable you can again either click on the square brackets surrounding the word or click on the word itself, and as before a menu may pop up if the word is spanned by more than one markable, where you first have to select the correct one. In the screenshot below the word "Raubes" will be chained together with "Totschlags".



[**Beteiligte**] an der [**Ermordung**] der [**deuts**
ger hat sich des [**Raubes**] und [**Totschlags**] s
ts
igen Xou Yang jetzt publik .

If a destination markable is already in a different chain you can either merge the chains belonging to each markable, or you can adopt the destination markable into the chain of the source markable. The destination markable will then be removed from the chain before, but the chain will still exist with its other members. Beware that if the old chain consisted of only two members, the chain will be deleted.



er dritte [**Beteiligte**] an der [**Ermordung**] der [**deutschen** [**Touristin**]] Gisela Pfleger hat sich des [**Raubes**] u
tschlags] schuldig bekannt .

ie [**Staatsanwaltschaft**] in Riverside in Kalifornien machte das [**Bekenntnis**] des 20jährigen Xou Yang jetzt pub

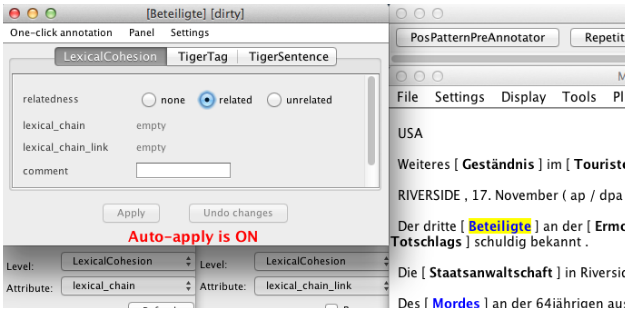
es [**Mordes**] an der 64jährigen aus Emmerich sind bereits der 19jährige Thongxay Nilakout und der 20jährige
amchan Bret

ie [**Strafmaß**

ı [**lebenslangen** [**Freiheitsstrafen**]] wurden zwei [**junge** [**US-Amerikaner**]] in Miami wegen des [**Mordes**] aı
utschen [**Touristen**]] Uwe-Wilhelm Rakebrand verurteilt .

🟦 adopt item into current chain
🟦 merge item (and dependents) into current chain
🟥 link item

If you encounter some markable that is relevant for the context but does not suit into any chain or link, check the "related" option on the markable attribute window on the top left (see below). Another option you may want to check is "unrelated" which switches the color of the markable in the display. This can be useful if a markable is visually too prominent and mainly disturbs the flow. Either way, selecting the option "none" or "unrelated" makes no difference in the final result.



[Beteiligte] [dirty]

relatedness ○ none ● related ○ unrelated
lexical_chain empty
lexical_chain_link empty
comment

Auto-apply is ON

Level: LexicalCohesion
Attribute: lexical_chain

Level: LexicalCohesion
Attribute: lexical_chain_link

USA

Weiteres [**Geständnis**] im [**Touriste**

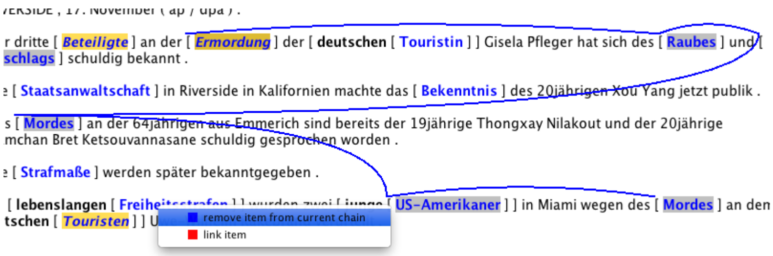
RIVERSIDE , 17. November (ap / dpa

Der dritte [**Beteiligte**] an der [**Ermc**
Totschlags] schuldig bekannt .

Die [**Staatsanwaltschaft**] in Riversic

Des [**Mordes**] an der 64jährigen au:

If you want to correct a decision, i.e. you want to remove a markable from a chain or remove a link between markables, you first have to select a source markable (any other markable in the chain, or the source markable of the link) and then right click on the desired markable you want to correct. A menu pops up where you select the desired action. See below for an example.



ERSIDE , 17. November (ap / dpa) .

r dritte [Beteiligte] an der [Ermordung] der [deutschen [Touristin]] Gisela Pfleger hat sich des [Raubes] und [schlags] schuldig bekannt .

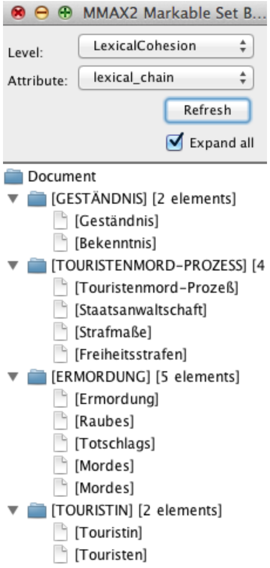
e [Staatsanwaltschaft] in Riverside in Kalifornien machte das [Bekenntnis] des 20jährigen Xou Yang jetzt publik .

s [Mordes] an der 64jährigen aus Emmerich sind bereits der 19jährige Thongxay Nilakout und der 20jährige mchan Bret Ketsouvannasane schuldig gesprochen worden .

e [Strafmaße] werden später bekanntgegeben .

[lebenslangen [Freiheitsstrafen]] wurden zwei [junge [US-Amerikaner]] in Miami wegen des [Mordes] an den tschen [Touristen]] U

On the lower left corner a so called markable set browser can be found, showing the chains in a list fashion. Clicking on an entry here will select the corresponding markable in the main display (see below). Check “Expand all” and click “Refresh” to view all list entries.



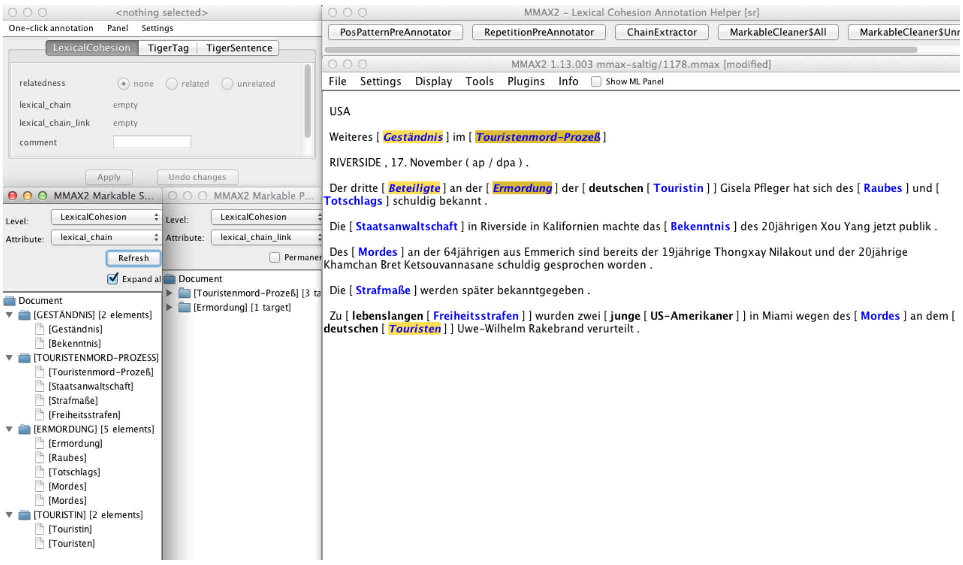
Level: LexicalCohesion

Attribute: lexical_chain

☑ Expand all

Document
- [GESTÄNDNIS] [2 elements]
  - [Geständnis]
  - [Bekenntnis]
- [TOURISTENMORD-PROZESS] [4
  - [Touristenmord-Prozeß]
  - [Staatsanwaltschaft]
  - [Strafmaße]
  - [Freiheitsstrafen]
- [ERMORDUNG] [5 elements]
  - [Ermordung]
  - [Raubes]
  - [Totschlags]
  - [Mordes]
  - [Mordes]
- [TOURISTIN] [2 elements]
  - [Touristin]
  - [Touristen]

Finally, save the annotations (“File -> Save -> All”) and export the chains and links by clicking the button “ChainExtractor” on the top.



relatedness ● none ○ related ○ unrelated

lexical_chain empty

lexical_chain_link empty

comment

Level: LexicalCohesion

Attribute: lexical_chain

☑ Expand al

- Document
  - [GESTÄNDNIS] [2 elements]
    - [Geständnis]
    - [Bekenntnis]
  - [TOURISTENMORD-PROZESS]
    - [Touristenmord-Prozeß]
    - [Staatsanwaltschaft]
    - [Strafmaße]
    - [Freiheitsstrafen]
  - [ERMORDUNG] [5 elements]
    - [Ermordung]
    - [Raubes]
    - [Totschlags]
    - [Mordes]
    - [Mordes]
  - [TOURISTIN] [2 elements]
    - [Touristin]
    - [Touristen]

Level: LexicalCohesion

Attribute: lexical_chain_link

□ Permaner

- Document
  - [Touristenmord-Prozeß] [3 ta
  - [Ermordung] [1 target]

USA

Weiteres [*Geständnis*] im [*Touristenmord-Prozeß*]

RIVERSIDE , 17. November (ap / dpa) .

Der dritte [*Beteiligte*] an der [*Ermordung*] der [**deutschen** [Touristin]] Gisela Pfleger hat sich des [Raubes] und [Totschlags] schuldig bekannt .

Die [Staatsanwaltschaft] in Riverside in Kalifornien machte das [Bekenntnis] des 20jährigen Xou Yang jetzt publik .

Des [Mordes] an der 64jährigen aus Emmerich sind bereits der 19jährige Thongxay Nilakout und der 20jährige Khamchan Bret Ketsouvannasane schuldig gesprochen worden .

Die [Strafmaße] werden später bekanntgegeben .

Zu [**lebenslangen** [Freiheitsstrafen]] wurden zwei [**junge** [**US-Amerikaner**]] in Miami wegen des [Mordes] an dem [**deutschen** [*Touristen*]] Uwe-Wilhelm Rakebrand verurteilt .

Have Fun!