Modeling the relevance of sentences

Detecting domain specificity and semantic relatedness from scientific texts

Henri Leisma

12.09.2011

University of Eastern Finland
School of Computing
Master’s thesis

Abstract

This thesis discusses the sub-problems in discourse segmentation of texts and presents a latent features approach to improving sentence topic modeling. Both lexical and semantic features are considered in the modeling. The goal is to find small sets of essential terms needed in relating the sentence to domain and world knowledge of the reader.

A hybrid approach using both linguistic resources and local distributional measures is used. WordNet is used as a filter and classifier in the processing. First a highly domain-specific vocabulary is collected from the text by excluding all terms appearing in the external resources. The vocabulary is then used in finding terms co-occurring with the highly domain-specific terms.

Semantic relatedness detection is attempted based on local sentence features and external resource hierarchical relation discovery. The extracted sentence latent features are used in constructing a topic signature for each sentence. The relevance of a sentence to the whole document is determined by categorization.

The resulting relevance model can be used for example in improving document indexing and question answering systems. The presented approach in the experimental part is meant as a proof of concept and the model can be improved by processing a large number of documents from the same domain.

ACM-classes (ACM Computing Classification System, 1998 version): I.7.2, I.2.7, I.2.4, I.5.1

Keywords: text summarization, topic signatures, semantic smoothing, topic detection, context analysis, feature extraction, semantic indexing, text mining, natural language processing

Contents

1 Introduction 1

2 Topic extraction subtasks 7

2.1 Text segmentation 7

2.2 Topic signatures 10

2.3 Semantic smoothing 15

2.4 Chaining the subtasks together 17

3 Extracting semantic features from text 19

3.1 Natural Language Processing 19

3.1.1 Part-Of-Speech Tagging 20

3.1.2 Dependency Parsing 21

3.1.3 Lemmatization 22

3.1.4 Word Sense Disambiguation and topic modeling 23

3.2 Resources for computing domain specificity and semantic relatedness 24

3.2.1 Dictionaries and thesauri 24

3.2.2 WordNet 27

3.2.3 FrameNet 28

3.2.4 Ontologies 31

4 Relevance and semantic relatedness measures 33

4.1 Domain vocabularies as relevance measures 33

4.2 Semantic similarity vs. semantic distance 34

4.3 Methods for computing semantic relatedness 36

5 Proposed relevance modeling algorithm 42

5.1 Constructing topic signatures 43

5.2 Computing domain specificity 44

5.3 Computing semantic relatedness 46

5.4 Categorization to relevant and irrelevant 48

6 Case study: Extractive summary generation 51

6.1 Methods and input data 51

6.2 Results and discussion 53

7 Conclusions and future work 57

References **59**

Attachment 1: Algorithm input data (testdoc1) **65**

Attachment 2: Categorization results (testdoc1) **68**

Attachment 3: Generated extractive summaries **71**

Attachment 3: Precision and recall of extractive summarization **74**

1 Introduction

Written natural language does not consist of isolated, unrelated sentences, but instead from structured and coherent groups of sentences. This is commonly referred to as *discourse*. There are two main types of discourse: *monologue* and *dialogue*. Monologues are unidirectional and characterized by a writer communicating to a reader. This is the type of discourse where written texts fall into. (Jurafsky and Martin, 2009)

The task of discourse segmentation depends on finding *discourse markers*. Discourse markers are used in linking *discourse segments* like sentences or clauses and thus defining cohesion boundaries in the text. A common approach to finding lexical *cohesion* is detecting *connectives*, which are fragments of text that are known to signal discourse structure. These can be cue words or cue phrases like "joining us next is (PERSON)".

Scientific texts, like any strictly factual texts, are expected to have high cohesion. Indicators of high cohesion are that the way textual units are linked together is consistent and references between discourse segments are unambiguous. *Coherence*, consistency in the meaning relation between two units, is another important feature. But detecting meaning relations requires semantic knowledge. Semantics can be divided to *context knowledge* of the current situation or domain and to broader *world knowledge* of the un-written human expectations of how things work and what they are related to. This is a key difference between human and computer interpretation of a text; humans reflect a written sentence to their personal world knowledge and to the domain-specific knowledge when reading. For a computer to be able to find related terms as well as a human, it would also need world knowledge to find the underlying relations. And to be able to select only the important relations for the situation, it would need knowledge of the current use context.

Terminology and key approaches

In the field of information retrieval (IR), the generic concept of *document* stands for a unit of text indexed and available for retrieval (Jurafsky and Martin, 2009). In principle, a document can refer to anything from full text documents to smaller units like paragraphs and sentences. A *collection* then refers to a set of documents. Documents consist of smaller fragments of text, commonly referred to as *terms*. Terms can refer to a lexical item (word or detected *named entity*) but also to phrases of text. When categorizing segments, a *topic* can be used as a signifier of a segment (Manning, 1998).

Documents and collections can be searched by constructing and executing a *query*. Named entity recognition is not included in the experimental part of this thesis. Key terms are summarized in Table 1.

Key term	***Short description***
discourse	the whole text, the complex concept being discussed
document	a sub-segment of the text (can be a clause, a full sentence, a paragraph, the entire document)
topic signature	a head topic of a document, a set of core terms and their relations describing the document
discourse marker	a detected boundary in the text, marking a change in topic
reference	a pointer (usually backward) to another sentence or concept

Table 1: Key terms in discourse segmentation and their short descriptions.

Splitting text and detecting the content of the segments may seem straightforward, which it is for an educated human being. But from a computer's point of view, there are several unsolved sub-problems. Handling semantics is particularly problematic, since the words of individual sentences seldom contain all the information needed to understand the meaning. Scientific texts make the problem even harder, since they have a highly domain-specific vocabulary and require broad contextual knowledge from the reader. Common problems are related to *reference resolution*, which tends to still require manually annotated data (Miltsakaki and Kukich, 2000). Reference resolution means solving which entities are referred by which linguistic expressions. There are two main classes of reference resolution tasks. The first task, *pronominal anaphora resolution*, handles single pronoun antecedent references like where in the previous sentences does the word he/she/it refer to. The second, a lot harder problem, is called *co-reference resolution*. It aims at finding all the entity referring expressions in the given context and grouping them into co-reference chains. An example of co-reference resolution would be detecting that “vehicle” and “car” refer to the same entity.

Another problem is resolving *coherence relations* between the encountered terms. Sentences in text are considered coherent, when the reader can form a semantic connection between them (Jurafsky and Martin, 2009). For example, a text where first you tell that a car accident occurred, would be coherent if you continued by giving a cause like the driver was drunk or the road was icy. A special case of finding coherence from text is detecting *entity-based coherence*. This means looking at not only direct references between individual terms but first using *named entity recognition* algorithm in extracting the entities and then finding references to these. An example of this would be detecting that “Bill Clinton” and “president” refer to the same entity.

Hierarchical relations between the terms play a key role in constructing a *context knowledge* model from the text. Terms with similar meanings have three hierarchical levels: higher in the hierarchy (*hypernym*), lower (*hyponym*) or on the same level (*synonym*). A key problem in this type of semantic detection is determining whether the terms have similar meaning or related meaning. Sometimes synonyms can be seen as *parallel concepts* and how fine-grained distinguishing should be used is reader dependent. The same word can also have multiple different meanings (*polysemy*), depending on the usage context. Finding the right meaning is called *word sense disambiguation* (WSD). These are all problematic for a computer, because, unlike a human interpreting a text, a computer does not have contextual knowledge or *world knowledge* at hand.

In text summarization, a closely related field to discourse segmentation, there are two key approaches to text processing. In the traditional approach only *lexical features* are taken into account. These can all be computed locally, that is when processing the text for the first time. Another approach is to take advantage of *structural features*. This is closer to the human approach, but requires processing the whole text beforehand, and is thus computationally more expensive. The use of world knowledge, reflecting encountered information to a conceptual model of all things known by the interpreter, would in turn require even more preprocessing and modeling. No computer-generated, human world knowledge equivalent conceptual models exist at the time of writing this thesis. Also many sub-problems of constructing such a model, like co-reference resolution and semantic role labeling, are still unsolved research problems.

Three layers of linguistic knowledge

Putnam (1975) distinguishes three layers of linguistic knowledge: 1) *domain vocabularies*, 2) *wordnets* and 3) *central ontology*. The three layers are distinguished according to his *principle of the division of linguistic labor*. Such a division is required to handle and structure the large quantities of domain vocabulary and its linguistic diversity.

The Blackwell Dictionary of Western Philosophy (Bunnin and Jiyuan, 2004) explains the connections of Putnam's principle to *social context* and *domain-specific knowledge*. According to it, language is used in a community, and a community is divided into many subsets. A word in a language may have different meanings and extensions, depending on its different references and the occasions on which it is used. The expert speakers may know all facets of the word and be aware of its various distinctions, but this will not be the case for average speakers. Not all of them can know all the distinctions or the exact extension. They use the word in the way that is accepted by the subset of the community to which they belong. By virtue of this principle, Putnam tries to indicate that not every term is a description, and that the extension of each term is at least partly determined socially rather than in the mind of the individual speaker.

- *Domain vocabularies:* Words collected from the text at hand.
- *Wordnets:* Words and their relations collected from a large corpus of texts.
- *Central ontology:* Mental model of high-level concepts and their relations to words.

In this thesis the focus is on constructing domain vocabularies and using wordnets as supporting resource in finding relations between domain words.

Five types of lexical cohesion

Halliday and Hasan (1976) laid the foundation for *lexical chains* by studying *lexical cohesion* relations. They suggested relating words of a text back to the first word to which they are cohesively tied. They also specified five types of lexical cohesion, based on the dependency relationship between the words.

- *Reiteration with identity of reference:* The nurse fetched a clean needle. It looked sharp, that needle.
  (a needle <– that needle)
- *Reiteration without identity of reference:* The doctor looked at his stethoscope. In his opinion, there could be no better stethoscope.
  (his stethoscope <– a stethoscope)
- *Reiteration by means of superordinate:* The nurse picked a fresh needle. She was comfortable with handling pointy instruments.
  (a needle <– pointy instruments)

- *Systematic semantic relation (systematically classifiable):* The lines on the floor leading to surgery were colored red. To the morgue, green.
  (red lines <– green lines)

- *Nonsystematic semantic relation (not systematically classifiable):* Jenny was the most experienced nurse. She was always in charge of new vaccinations.
  (nurse <– nurses handle vaccinations)

In the first example there is an identifying reference (referer: that needle) and in my opinion this type is the easiest to detect algorithmically. The ambiguity of the reference increases with each example and thus also the complexity of resolving the reference. The second reference example is also clear in the way that it repeats the earlier encountered word. It does not identify a specific target though (referer: a stethoscope). Resolving the reference in the remaining three examples requires prior knowledge of the world. In example three, the hierarchical relation between a needle and pointy instruments needs to be known. In example four, the immediate context needs to be temporarily constructed (that the second sentence talks about lines and that green is an attribute value of line). The last example is the hardest, requiring knowledge of actions and attributes related to the entity being indirectly referred to (nurses handle vaccinations).

Lexical chains, a segmentation method that uses lexical cohesion, are discussed in more detail in section 4.2.

Related fields of computer science

Detecting topics from a sentence, like many other complex problems in computer science, includes a set of subtasks. There are several sub fields of computer science like computational linguistics, natural language processing, pattern recognition, clustering, knowledge representation and information extraction to name a few. Some understanding of linguistics and statistics is also required. Linguistics provide most of the terminology used in lexical processing, semantics related terms and models mostly from psychology. Methods from statistics are used in extraction and analysis of text features.

One of the key fields is *natural language processing* (NLP), which provides, for example, *part-of-speech tagging* (POS) and *dependency parsing*. Most modern NLP algorithms are based on statistical machine learning. This means that, rather than hand-coding a large set of rules, statistical inference is used in learning such rules. This is achieved by first analyzing a large corpora of examples and constructing a statistical model from this *training set*. The learned statistical model is then used in predicting where in the model a new input would best fit. Many of the tasks in NLP serve as subtasks that are used to aid in solving larger tasks.

Data mining and information extraction are used in domain acquisition from texts. Data mining is mostly applied to finding relations from external sources whereas information extraction is used for constructing data sources from the text itself. Categorization and clustering methods are commonly used when grouping text fragments and finding similarities.

Related work on discovering hierarchical relations

There are many related works on discovering hypernyms from text, most of which follow the methods of Hearst (1992). Kennedy and Szpakowicz (2007), who worked on thesaurus data, mention that it is time-consuming to construct a large lexical resource that would be as trustworthy as WordNet (Fellbaum, 1998), therefore much work is still left to be done manually. Hearst (1992) was the first to create hypernym hierarchies automatically from a corpus.

Nakamura and Nagao (1988) mined dictionaries for relations already in the eighties and included relations other than hypernyms. Jarmasz and Szpakowicz (2003a, 2003b) have used Roget's thesaurus in constructing lexical chains and detecting semantic similarity. Later Kennedy and Szpakowicz (2007) worked on disambiguating some of the hypernym relations in the thesaurus data. bag-of-words approach Agirre *et al.* (2001) worked with topic signatures as a tool in enriching WordNet concepts. Their work was motivated by the lack of topical links among concepts in WordNet. WWW and sense-tagged corpora were used as a source for finding new relations.

Pantel and Ravichandran (2004) have conducted research on labeling semantic classes using IS-A relations. Snow *et al.* (2005) introduced machine learning approaches in the identification of hypernyms in text. More advanced systems, such as Espresso (Pantel and Pennacchiotti, 2006), have been designed to identify also other semantic relations from text.

2 Topic extraction subtasks

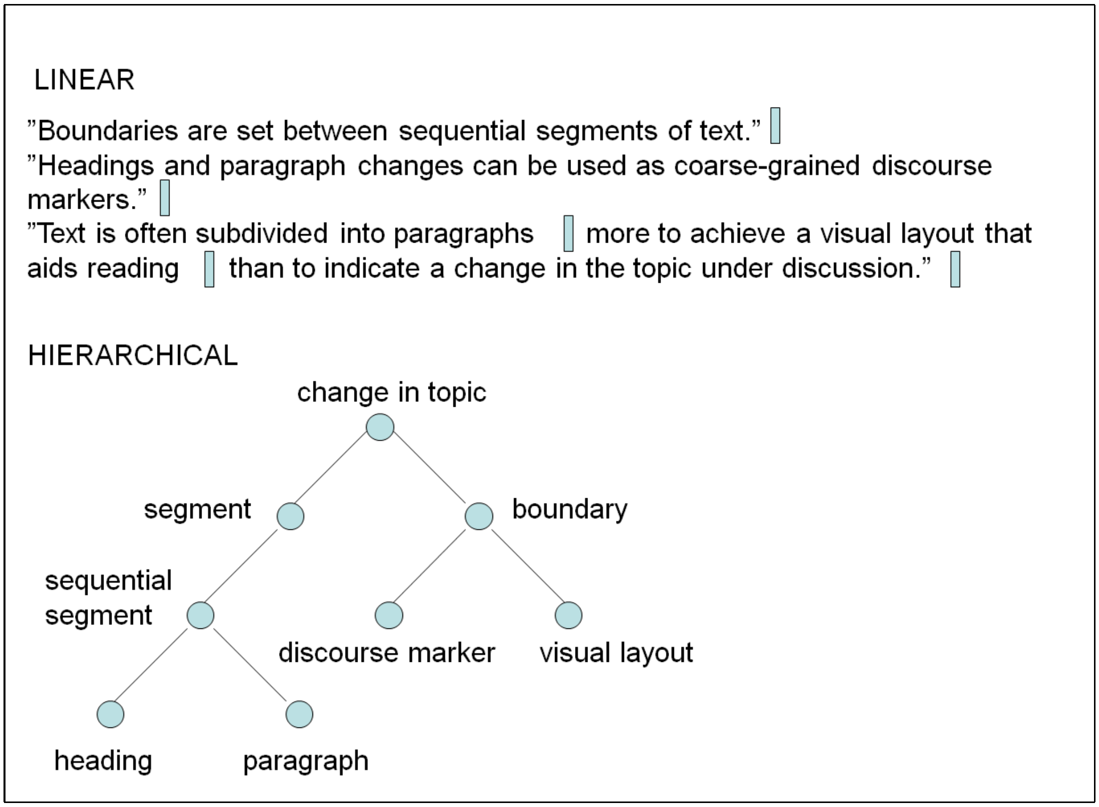
Discourse segmentation is about splitting a text into smaller segments. Topic detection for each sentence is a key subtask in discourse segmentation. The task is started by splitting the source text into segments (for which we want to detect the topics). For example, splitting a scientific paper (a discourse itself, but a broader one that we are after) to a set of sentences (collection of documents). For representing the topics, we need to construct at least one *topic signature* for each sentence. Topic signatures contain selected sentence components, their attributes and relations. The signatures can be improved by *semantic smoothing*, by finding related terms from external resources and including them in the signature. This binds the signature to the underlying broader context of the text.

2.1 Text segmentation

Manning (1998) describes *discourse structure* as a set of coherent units extracted from a text, often represented as a discourse structure tree. The expected segmentation result is that coherent units of discourse can be detected and that they will describe a single event and can thus form a sub-tree of the entire discourse tree.

Text segmentation (also referred to as *discourse segmentation*) is the task of dividing a text document into cohesive segments by topic (Hollingsworth, 2008). There are two main approaches to discourse segmentation, *linear segmentation* and *hierarchical segmentation* (Manning, 1998). Figure 1 summarizes the key differences between linear and hierarchical approaches. In linear segmentation, boundaries are set between sequential segments of text and the detected segments are not further subdivided. In hierarchical segmentation, in turn, each detected segment can be further divided into sub-segments and linked to other related segments. Scientific texts always contain sections and sections contain subsections and the text itself is further divided into paragraphs. This visual structuring can act as a coarse-grained discourse segmentation (Kawtrakul and Yingsaeree, 2005), where headings and paragraph changes are used as *discourse markers*. In this thesis, the focus is on mapping sentences to the underlying semantic discourse structure. Stark (1988) observed that *real world text* (as opposed to sequences of computer-generated sentences) is often subdivided into paragraphs more to achieve a visual layout that aids reading than to indicate a change in the topic under discussion.

Text segmentation is needed as a subtask in many computational linguistics tasks, for example in text summarization and IR. Text segmentation is often the first step in *extractive text summarization* (Barzilay and Elhadad, 1997), in which a summary is constructed by choosing sentences from the text itself. In IR, Salton et al. (1993) have found that comparing a query against sections and then paragraphs yields more relevant search results than comparing only against entire documents. Users find it also more helpful if the relevant paragraph(s) are displayed in the results of their query (Hearst, 1997).



LINEAR

”Boundaries are set between sequential segments of text.” ▯
”Headings and paragraph changes can be used as coarse-grained discourse markers.” ▯
”Text is often subdivided into paragraphs ▯ more to achieve a visual layout that aids reading ▯ than to indicate a change in the topic under discussion.” ▯

Figure 1: Linear and hierarchical segmentation. Linear segmentation uses only sequential boundary markers where hierarchical segmentation attempts to construct a concept tree. A tree structure is useful in calculating term relatedness values between components of different sentences.

Linear approach

A simplification of the linear approach is to say it is just taking a knife and cutting the text to pieces. Grosz and Sidner (1986) criticize this approach, because within theoretical work on discourse structure, it is standardly assumed that discourse has a hierarchical tree structure. An attempt to induce such hierarchical structure from text is presented in (Morris and Hirst, 1991).

In empirical work on text segmentation, which attempts to automatically label discourse structure, the assumption of hierarchical nature has largely been abandoned, and discourse structuring is seen as merely a task of linear segmentation. Hearst (1994) notes that the hierarchical view of discourse is standard, but suggests that a linear segmentation is sufficient for some domains of interest. Examples of these include finding segments for use in WSD and limiting search and returning context in IR systems.

Hierarchical approach

The hierarchical approach is much harder than the linear approach, since it requires identifying non-sequential relations. This is difficult and time-consuming even for human annotators (Passonneau and Litman, 1993). Passonneau and Litman mention also that non-linear segmentation is impractical for naive subjects in discourses longer than 200 words. Although some tasks in text segmentation can be performed with the simpler linear approach, others like co-reference resolution depend completely on the recognition of hierarchical discourse structure (Manning, 1998).

The two basic techniques that have been used for segmentation are *cue phrases* (Grosz and Sidner, 1986; Passonneau and Litman, 1993) and *lexical cohesion* (Morris and Hirst, 1991; Hearst, 1994). *True cue phrases* (such as "for example" or "basically") are valuable in processing both spoken and written discourse. Manning (1998) states that cue phrases are not much use in segmenting naive texts like housing advertisements, but still certain elements such as suburbs and prices could be used as generalized cue phrases.

Lexical chains in text segmentation

Morris and Hirst (1991) were the first ones to use lexical chains for discourse segmentation. This approach has since become a standard application of lexical chains. Lexical chains consist of semantically related words, each chain corresponds to a theme or topic (or a set thereof) in the text (Morris and Hirst, 1991). Figure 2 explains the use of lexical chain overlap in merging chains.

The chains have beginning points and ending points, which mark the *chain boundaries* of that chain. Lexical chains provide at least the following three useful cues. These

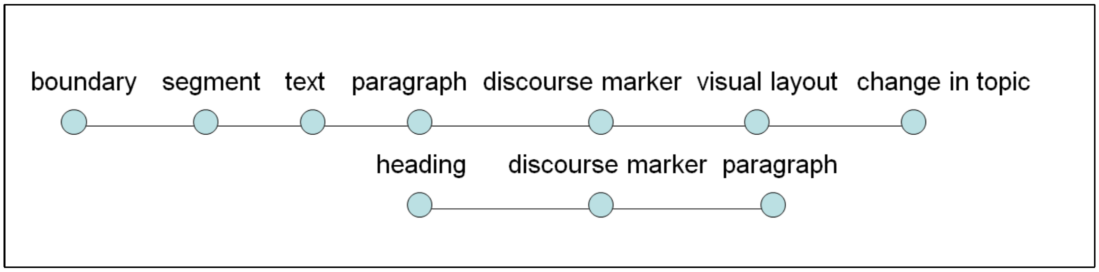


Figure 2: Lexical chain overlap. New chains start from the new terms, which alone cannot be connected to existing chains. By merging new chains to existing chains, longer chains can be produced and thus a more coarse-grained topic segmentation achieved.

cues can help in detecting positions at which there are shifts in topic, representing *topic boundaries*.

- *Chain beginning points:* A significant number of chains beginning at a point in text probably indicates the emergence of some new topic(s).
- *Chain ending points:* A significant number of chains ending at a point in text probably means that certain topics are not discussed henceforth in the text.
- *Low number of chain points:* Points where the number of chains beginning or ending is not significant probably represent a continuation in the discussion of some topic(s).

2.2 Topic signatures

Topic signatures are sets of related words, with associated weights, organized around *head topics*. Head topics are much like headings in text, a few keywords describing the contained text. Once a text has been split into segments, each segment should contain only words that are related. Based on the contained words and their relations the segment can be named. In linear segmentation, words from sequential sentences form sets of related words, whereas in hierarchical segmentation the sets are sub-trees of the whole discourse tree.

Topic signatures are a useful tool in automated text summarization. They can be used to identify the presence of a *complex concept* - a concept that consists of several *related components* in *fixed relationships* (Lin and Hovy, 2000). `Restaurant-visit`, for example, involves at least the concepts `menu`, `eat`, `pay`, and possibly `waiter`. Only when sufficiently many of the concepts co-occur, one can infer the complex concept; eating or paying alone are not sufficient. The presented example does not yet consider the interrelationships among the component concepts, which is an important context distinguishing factor in topic signature inference. Lin and Hovy (2000) point out that many texts describe all the components of a complex concept without ever explicitly mentioning the *underlying complex concept*, a topic itself. Because of this, systems that have to identify topics require a method of inferring complex concepts from their component words in the text.

Defining topic boundaries

A naive approach to defining topic boundaries in text is extracting them only from the structure of text. However, a single sentence or paragraph rarely contains the entire discussion on one topic. For example, in this section, we have written several paragraphs about topic signatures and will return to their practical detection aspects in the later sections. A broader perspective is needed in defining topics and their boundaries.

The two common approaches to detecting topic boundaries are the dynamic *tracking approach* and the static *context approach*. Hearst (1994) describes the tracking approach as a relatively large set of active themes that change simultaneously. Kozima (1993), in turn, suggests visualizing the context approach as a scene in a movie which describes the same objects in the same situation. Which approach to choose depends on what the desired segments are. Figure 3 gives an example of the tracking approach and Figure 4 of the context approach. Beeferman *et al.* (1997) state that much of the existing literature on text segmentation is somewhat vague in defining the desired segments. They adopt the empirical definition that a segment boundary is the article boundaries between news reports in newswire corpora. In segmenting scientific papers, this would mean that the entire paper defines the topic. Our goal is to find much more fine-grained subtopics from within a paper.

Using lexical cohesion works well in news reports domain, because successive stories are almost always about completely different topics. This contrasts with the actual formatting of a newspaper, where stories on the same topic are normally grouped. With scientific papers, each individual paper should also be distinguishably different, having its own contribution, from the other papers in the same domain. And there is a similarity in grouping as well; a list of conference papers tends to contain papers from

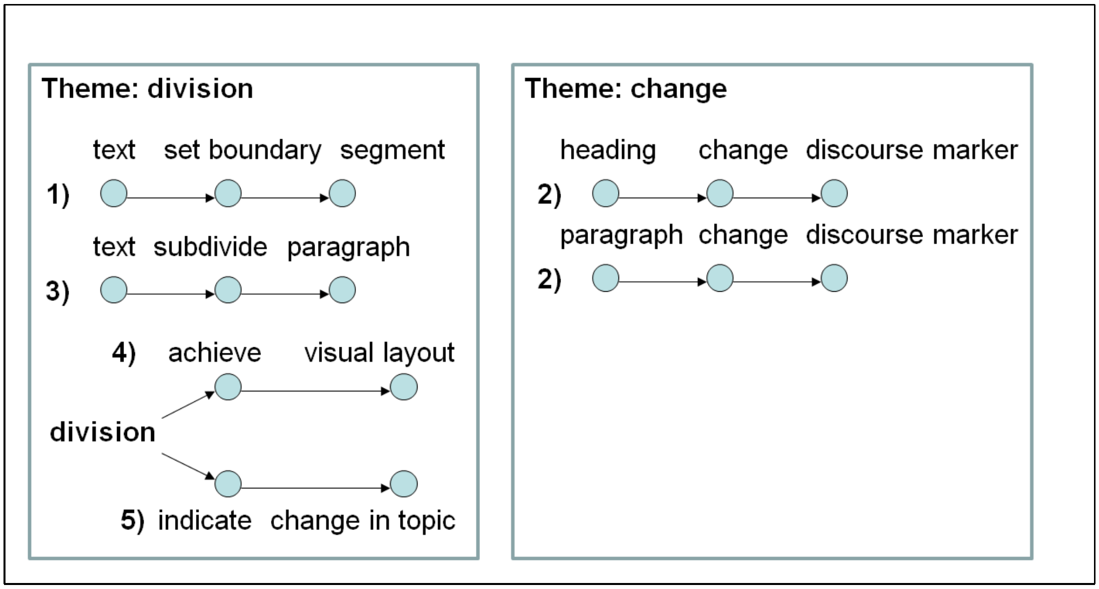


Figure 3: Tracking approach to detecting topic boundaries. The idea is to iterate through sentences and try to map sentence terms to an active theme. When this cannot be done, mark theme changes.

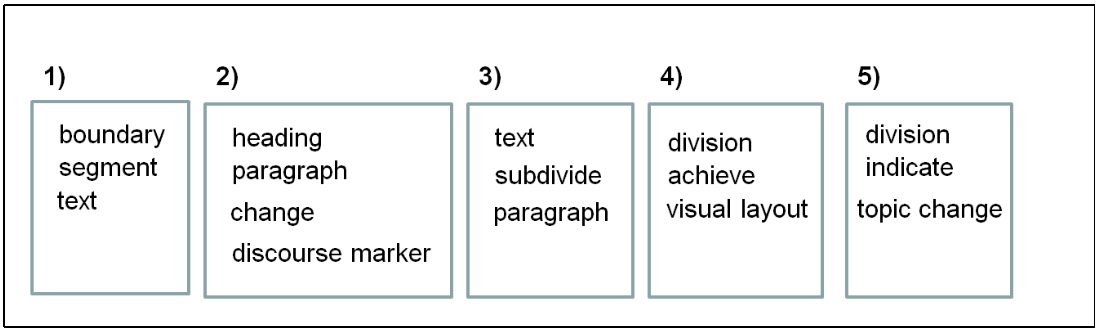


Figure 4: Context approach to detecting topic boundaries. The idea is to construct an independent context frame for each sentence and then iterate the frames to detect topic changes.

the same special interest group. There is, in fact, a tendency of this type of grouping in most text collections, which leads us to the problem of great lexical cohesion between independent texts appearing together. Beeferman (1997) gives an example of days when big news stories covering the entire front page may be devoted to one topic, but will contain multiple articles. Such circumstances cause great lexical cohesion between the individual articles on the page, but the articles emphasize different aspects of the same event and thus represent different “movie scenes” (Kozima, 1993). Similarly, within real estate ads, every ad talks about bedrooms, garages, locations and prices. There is great lexical cohesion throughout the entire real-estate section, but we still wish to segment it into smaller, unique descriptions of each property.

Naming a topic

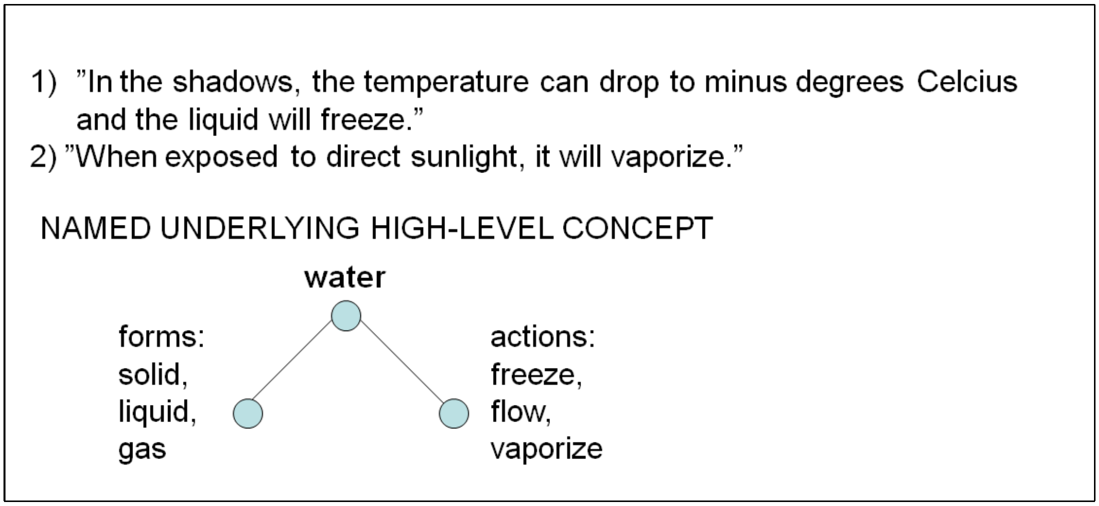
Why is naming a topic, giving it a signature, important in text segmentation? Consider unnamed topics, which do not have an easily referable, comparable or searchable signature. How would we construct a model from them without the ability to refer back to already detected concepts? There are three main approaches to naming a text topic: 1) using one of the contained terms 2) detecting the nearest common higher level concept and 3) constructing a name from the component concepts.

- *Contained term:* Selecting the highest ranked or most frequent contained term as the signature name. Requires weighting the terms.
- *Nearest hypernym:* Selecting the nearest hypernym of the highest ranked or most frequent contained term. Requires an external source of hypernyms like a thesaurus or wordnet.
- *Constructed name:* Finding several key terms and finding a way to combine them. Possible combining strategies include finding an example use, a template use case, where the terms occur together. Another approach is constructing a grammatically acceptable signature name based on the parts of speech of the terms, or when POS is not available, their order of appearance in the source text.

One of the key characteristics of a signature name is that a similar name should be derivable from the terms and relations belonging to the signature. The reason for constructing signature names is to be able to compare them to other signatures and to give a common name for the contained group of concepts. An example of a named underlying concept is given in Figure 5. When constructing signature names only from terms in the source text, there is a risk of *overspecification*, the signature becoming too bound to the appearance context. This reduces comparability to signatures describing the same concept, but which have been constructed from similar terms of another text. The opposite of overspecification is *underspecification*, a situation when an occurrence of terms results in a too generalized signature name.

Topic signatures in text segmentation

The central idea of systems based on lexical cohesion (Morris and Hirst, 1991) is that



1) ”In the shadows, the temperature can drop to minus degrees Celcius and the liquid will freeze.”

2) ”When exposed to direct sunlight, it will vaporize.”

Figure 5: A named underlying high level concept, the component concepts and the source text.

if the text continues to use similar words, then it is probably still talking about the same topic. It is important to note the difference between detecting *similar words* and searching *related words*. This type of simple looking for repeated words is insufficient, because describing the underlying concepts to a reader already familiar with the general topic does not require mentioning any of the higher level concepts of the current discourse. Most systems, such as (Morris and Hirst, 1991), use thesauri or semantic nets to enable evaluating cohesion at a semantic level, rather than at a lexical level. For hierarchical topic modeling, hierarchical synonyms of words can be found with the help of WordNet (Fellbaum, 1998).

General lexical cohesion will pick out a large unit on a certain topic, whereas in many cases one wants to separate out smaller units within that topic. The heuristic of grouping segments that are lexically or semantically similar is opposed to the most common heuristic for segmentation in information extraction (IE) systems. A common approach in IE systems is to extract individual slots and fillers, as shown in Table 2, and then to merge information into frames (Manning, 1998). This is based on using the high level heuristic that one merges information unless it is contradictory. For example, having extracted a place followed by a date, then one combines them as two components of information about one event. On the other hand, if one has extracted two places in turn, then they probably describe different events. This heuristic involves seeing *semantically cohesive words* as an indicator of a different event rather than a continuation of the same topic.

In many cases simply linking two words or sentences as related is not enough. There is a need to specify the type of relation and possibly define contextual restrictions to it. Specifying the relation type can be done as an attribute of the relation, but binding the relation to a context, to multiple other concepts and their relations, requires a different approach. Unlike most other text segmentation work, Beeferman *et al.* (1997) have modeled both *attraction* and *repulsion* between words as relation attributes, and their model can therefore generalize over both these intuitions. Attraction and repulsion attributes can be thought of as collections of categorized and weighted factors contributing to positive or negative correlation. This approach enables postponing total relation weight calculation and makes it possible to sort or filter the factors according to their importance to the context at hand.

2.3 Semantic smoothing

A common method of *clustering* text documents is a *bag-of-words approach*, where each document is represented as a list of words or phrases, a *word vector*. A weakness of the bag-of-words approach is losing the occurrence order of words. Too common words, like articles ("a","the") and conjunctions ("and","or"), are often filtered out with *stop word lists*. The key words of a domain can then be calculated by collecting word vectors from different documents. A matrix of documents and terms with term occurrence counts as values can be constructed in order to detect documents with similar sets of terms. This is known as the vector space model. A later improvement to the model is latent semantic indexing (Deerwester *et al.*, 1988).

A document is often full of domain-independent general words, like "approach" and

Role slots	***Role slot content examples***
subject: verb: object:	"Semantic smoothing improves topic signatures." (semantic smoothing)-(improves)-(topic signatures)
resource: metadata field: metadata content:	"(Zhou et al., 2007)" (this document)-(refersTo)-(other document)

Table 2: Syntactic role slots and filler examples. Triples are an example of simple slots, which can express source, relation and target.

“result”, and short of domain-specific core words. Zhang *et al.* (2006) explain that this makes document clustering challenging, because the lack of distinct terms results in more similar *training sets* than with documents containing specific terms. Fortunately, general words are not a big problem with scientific texts, since they include a lot of highly domain-specific terms. The greater problem with a scientific text, from a topic modeling point of view, is that a writer expects some (often significant) domain-specific knowledge from the reader. This means that many common concepts in the domain are not explained or even mentioned in the text.

Since a text does not contain all the terms needed to minimally describe the underlying broader context, finding terms and relations from external resources is required to fill the gap. This can be done by enriching each training set (terms collected from a document) by adding related terms collected from external resources like WordNet to that training set. Also attributes for the related terms can be calculated, for term weighting purposes, like a semantic relatedness value to the document terms. Document clustering experiments by, for example, Zhang *et al.* (2006) show that model-based clustering approach with *semantic smoothing* improves cluster quality.

Semantic smoothing, incorporating synonym and sense information into the training sets, is an effective way to improve retrieval performance (Zhou et al., 2007). Zhou *et al.* (2007) mention that earlier semantic smoothing models such as the translation model have shown good experimental results. A *translation model* means mapping terms to either synonymous or similar terms. Translation models are, unfortunately, unable to incorporate contextual information. Zhou *et al.* (2007) proposed a context-sensitive semantic smoothing method that decomposes a document into a set of weighted context-sensitive topic signatures. This type of signatures can then be used in IR tasks, for example in mapping the signatures into query terms supplied by the user.

In the experimental part of this thesis, semantic smoothing is applied to sentence topic signatures. A simple topic signature can consist of a set of words from one sentence. When an external resource is available, this set can be enriched with detected related words from that resource. A weight can be calculated for the signature, with the help of a collected *domain vocabulary*, for example to approximate how much contribution the sentence contains. Zhou *et al.* (2007) used pre-defined context centroids and expectation maximization (EM) algorithm in clustering the topic signatures to contexts. When incorporating context information with calculated attributes to the signature, a more complex data structure than just a word list is needed. One approach to improving the signature is structuring it to hold verbs, nouns, adjectives and adverbs in separate subsets or a matrix of terms and their attributes. Having part-of-speech information available for each term eases semantic distance calculations with thesaurus resources, like WordNet.

Semantic smoothing is an important step in improving signature quality. Modeling *semantic cohesion* from only the written words of a scientific text is, intuitively, seldom possible. Having context and world knowledge (similar and related terms) available when detecting context has been shown to improve model quality (Boyd-Graber et al., 2007). Table 3 gives examples of four common semantic binding types.

Binding type	***Examples***
state	"frozen cooler water" and "running engine" refer to the internal state of a car
action	"driving on a highway" and "stopping to refuel" refer to direct actions related to a car
event	"overtaking a truck" and "getting a speeding ticket" refer to situational events related to driving a car
temporal	"I bought it ten years ago" is a temporal expression and "a two hour drive in rush hour" refers to a temporal event

Table 3: Modeling semantic cohesiveness with bindings to context. The idea is to add attribute slots (like state, action, event and temporal) to the topic signature template. While processing each sentence, try to find context terms to fill the slots.

2.4 Chaining the subtasks together

The goal of the experimental part of this thesis is to find essential, domain-specific content from a scientific text. The method to achieve this consists of pre-processing the text with NLP tools, calculating domain specificity and semantic relatedness values for sentences and finally constructing an extractive summary of the text. Additionally, the intermediate steps of the algorithm produce data which could be used in labeling and indexing the whole document.

The following list contains algorithm phases related to topic signature construction.

The whole algorithm is described in more detail in Section 5.

- *Split text to sentences:* Construct the segments for which topic signatures will be generated.
- *Pre-process with NLP tools:* Extract words from sentences, use POS and lemmatization to improve comparability of words.
- *Enrich common words:* Fetch direct hypernyms for each sentence word from WordNet and add them to sentence word vector.

3 Extracting semantic features from text

3.1 Natural Language Processing

Sentence topic modeling depends heavily on the most common tasks in NLP. NLP provides methods to, for example, splitting the text to sentences and words (sentence segmentation), POS, dependency parsing and lemmatization. Many of the tasks in NLP serve as subtasks that are used to aid in solving larger tasks. In the following, we discuss the steps in NLP in their pre-requisite order. Segmentation is needed before tagging can be performed, tagging individual words is needed before parsing the whole sentence or lemmatizing a word to the correct root form becomes possible.

Sentence segmentation

Sentence segmentation refers to dividing a text into its component sentences. In English and some other languages, using punctuation (`. ? !`) is a reasonable approximation. However, even in English this is not trivial due to the use of the full stop character (`.`) for abbreviations, which may or may not terminate a sentence. An example of non-terminating dot is "... as he arrived. Dr. Jones participated ...", where "Dr." is not its own sentence. A precompiled abbreviations map can help prevent incorrect assignment of sentence boundaries.

Splitting a text into sentences is the simplest form of sentence topic modeling. This is done by assigning a unique, unnamed topic to each sentence. Unfortunately, this does not result in a connected sentence topics model, because it ignores all relational information, like references between sentences and same terms used in different sentences. However, splitting a text into sentences, is a good starting point for *categorization* or *clustering*, of which categorization is used in the case study. Extracting words of all sentences takes us one step closer to useful topic modeling. A simple distributional context approach is to use the words in nearby sentences as such in constructing a *context vector* for the sentence.

> **An example of sentence word vectors** with sentences, word vectors and enriched word vectors.

```
S1: Purpose of use is detected with cue words like with.
```

```
S2: Detecting cue words helps in finding causality relations.

Word vectors
S1: [ purpose, use, be, detect, cue, word ]
S2: [ detect, cue, word, help, find, relation ]

Enriched word vectors
S1: [ purpose, use, be, detect, cue, word,
      goal, find, discover, clue, evidence ]
S2: [ detect, cue, word, help, find, relation,
      observe, find, discover, clue, evidence ]
```

Sentence word vectors can be enriched with related words detected from external resources like WordNet. A common approach is to add close hypernyms (more general concepts) to sentence word vector. After enriching the word vector, parts of the sentence content, for example "detect cue word", can be expressed with paraphrases like "find evidence", "discover evidence".

At least some of the following NLP methods are needed for extracting the hidden features of the words and sentences. Depending on the type of information that the methods process, this is called *lexical topic modeling* or *semantic topic modeling*.

3.1.1 Part-Of-Speech Tagging

In corpus linguistics, part-of-speech tagging, also known as *grammatical tagging* or *word-category disambiguation*, is the process of marking up the words in a text (corpus) as corresponding to a particular part of speech. This is done based on both its definition, as well as its context, i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph. A simplified form of this is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc. In computational linguistics, POS tagging is done using algorithms which associate discrete terms, as well as hidden parts of speech, in accordance with a set of descriptive tags (Manning and Shutze, 1999).

> **An example POS result**, where tags starting with "V" mean verb and tags starting with "N" mean noun. (from Stanford POS Tagger)

```
Purpose of use is detected with cue words like with.
```

```
Purpose/NNP of/IN use/NN is/VBZ detected/VBN with/IN
cue/NN words/NNS like/IN with/IN ./.
```

POS tagging can be used as such in helping sentence topic modeling. Since the same words can have multiple senses, knowing the part of speech of a word can be used in limiting the possible senses to only those appearing with the detected POS. Borrowing an approach from *distributional word sense disambiguation*, a sentence context vector can be generated from the nearby words and their POS tags and the POS-limited set of known word senses. POS also enables constructing multiple contexts for the sentence, one containing only actions (verb), another for subjects, objects and related entities (nouns) and one even for descriptive words appearing nearby (adjectives).

3.1.2 Dependency Parsing

A natural language *parser* is a program that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb. *Probabilistic parsers* use knowledge of language gained from hand-parsed sentences, with the goal of producing the most likely analysis of new sentences. These statistical parsers still make some mistakes, but commonly work rather well. State-of-the-art Stanford parser has an accuracy of over 90 percent (Klein and Manning, 2003). Their development was one of the biggest breakthroughs in NLP in the 1990s.

The Stanford lexicalized probabilistic parser, used in our Scientific Writing Assistant project (SWAN, http://cs.uef.fi/swan/) and the related experimental part of this thesis, implements a factored product model, with separate probabilistic context-free grammar (PCFG) phrase structure and lexical dependency experts, whose preferences are combined by efficient exact inference, using an *A* algorithm* (Hart et al., 1968). The software can also be used simply as an unlexicalized stochastic context-free grammar parser. Either of these yields a good performance statistical parsing system: "We’ve parsed at a rate of about 1,000,000 sentences a day by distributing the work over 6 dual core processor machines." (Klein and Manning, 2003).

An example parse result with noun phrases “purpose of use” and “cue words" detected. (from Stanford Parser)

```
Purpose of use is detected with cue words like with.

(ROOT
  (S
    (NP
      (NP (NNP Purpose))
      (PP (IN of)
        (NP (NN use))))
    (VP (VBZ is)
      (VP (VBN detected)
        (PP (IN with)
          (NP (NN cue) (NNS words)))
        (PP (IN like)
          (PP (IN with)))))
    (. .)))
```

Parsing provides useful information for topic modeling. Especially the grammatical roles of words appearing around a verb are helpful. For example, constructing subject-verb-object sequences (triples) for each sentence. Parsing also provides groupings to noun phrases and verb phrases. This reveals the deeper hidden structures of sentences and enables for example entity recognition.

3.1.3 Lemmatization

A *lemma* in morphology is the canonical form of a *lexeme*. Lexeme, in this context, refers to the set of all the forms that have the same meaning, and lemma refers to the particular form that is chosen by convention to represent the lexeme. In lexicography, this unit is usually also the citation form or headword by which it is indexed, in other words the word under which a set of related dictionary entries appear. Lemmas have special significance in highly inflected languages, where for example the use of post-positions varies the word forms. The process of determining the lemma for a given word is called lemmatization.

An example of lemmatization. The difference between stemming and lemmatization is that stemming is strict string matching which cuts predefined word endings whereas lemmatization aims at finding the dictionary form of a word. (from MorphAdorner English Lemmatizer)

```
Word:            relation

Lemma:           relation
Porter stem:     relat
Lancaster stem:  rel
```

From the topic modeling point of view, lemmatization enables storing the words in their shorter root form. This makes comparisons between different instances of the same word easier to implement. Without lemmatization, *edit distance* between instances of the same word would have to be calculated. The edit distance between two strings of characters is the number of operations required to transform one of them into the other. Having the same word written differently results in the word instances always having an edit distance greater than zero. For detecting sentence topics, we want same words occurring in different sentences to be detected as accurately as possible. For entire document indexing, on the other hand, this would be of lesser importance.

3.1.4 Word Sense Disambiguation and topic modeling

Many words have multiple meanings. The process of identifying the sense of a word in a particular context is known as word sense disambiguation. WSD is an important task in NLP, and is a key component in, for instance, machine translation and IR.

An example of word senses for one word. The example contains four senses of the word “detection”. (from WordNet)

```
* S: (n) detection, sensing
  (the perception that something has occurred ...)
  "early detection can often lead to a cure"
* S: (n) detection, catching, espial, spying, spotting
```

```
  (the act of detecting something; catching sight ...)
* S: (n) signal detection, detection
  (the detection that a signal is being received)
* S: (n) detection, detecting, detective work, sleuthing
  (a police investigation to determine the perpetrator)
  "detection is hard on the feet"
```

Recently researchers have experimented with topic models (Cai et al., 2007) for sense disambiguation and induction. *Topic models* are generative probabilistic models of text corpora in which each document is modeled as a mixture over (latent) topics, which are in turn represented by a distribution over words. Approaches using topic models for WSD either embed topic features in a supervised model (Cai et al., 2007) or rely heavily on the structure of hierarchical lexicons such as WordNet (Boyd-Graber et al., 2007).

One of the main goals of this thesis is to apply topic modeling techniques from WSD to discourse segmentation, specifically to sentence topic modeling. There are also many other advanced NLP techniques, like *named entity recognition* (NER) and *semantic role labeling* (SRL), which might also be applied to discourse segmentation. But attempting to develop algorithms also for these entire sub fields of NLP is out of the scope of this one thesis.

3.2 Resources for computing domain specificity and semantic relatedness

There are four types of lexical resources for computing *semantic distance*. A common approach is to compute semantic distance from the lexicographers' judgments that are implicit in dictionaries, thesauri, semantic nets and WordNets (Fellbaum, 1998) or FrameNet (Fillmore et. al, 2010).

3.2.1 Dictionaries and thesauri

A thesaurus helps to find a word that is wanted and a dictionary defines it. One of the commonly used dictionaries is the *Longman Dictionary of Contemporary English*, which is available also online (http://www.ldoceonline.com/). It consists of headwords and has it's own defining vocabulary. In general, dictionaries are described as closed *paraphrasing* systems of natural language, which means restatement of a text or it's semantic content using other words. A paraphrase typically explains or clarifies the text that is being paraphrased.

Dictionaries

One of the dictionary-based approaches to computing semantic distance includes the so-called Kozima and Furugori's *spreading activation algorithm* (Kozima and Furugori, 1993). The idea, in brief, is to create a node for every headword and link this node to the nodes corresponding to all the headwords in its definition. When a word is encountered, it triggers processing of that words' dictionary relations as well. Kozima and Ito have later improved this algorithm by introducing adaptive scaling (Kozima and Ito, 1997). The difference of these two approaches is to treat context-free (static) and context-sensitive (dynamic) distances differently. A characteristic of such straightforward dictionary approaches is to ignore the context and focus on direct mappings between words. When context is considered, the perceived distance changes with the observed context. Associations constructed from the dictionary usually have a direction in the mapping (engine is part of a car and bus is a type of vehicle). It is difficult for a computer to detect hierarchical relation from a textual dictionary definition unless it is explicitly given (Kozima and Furugori, 1993).

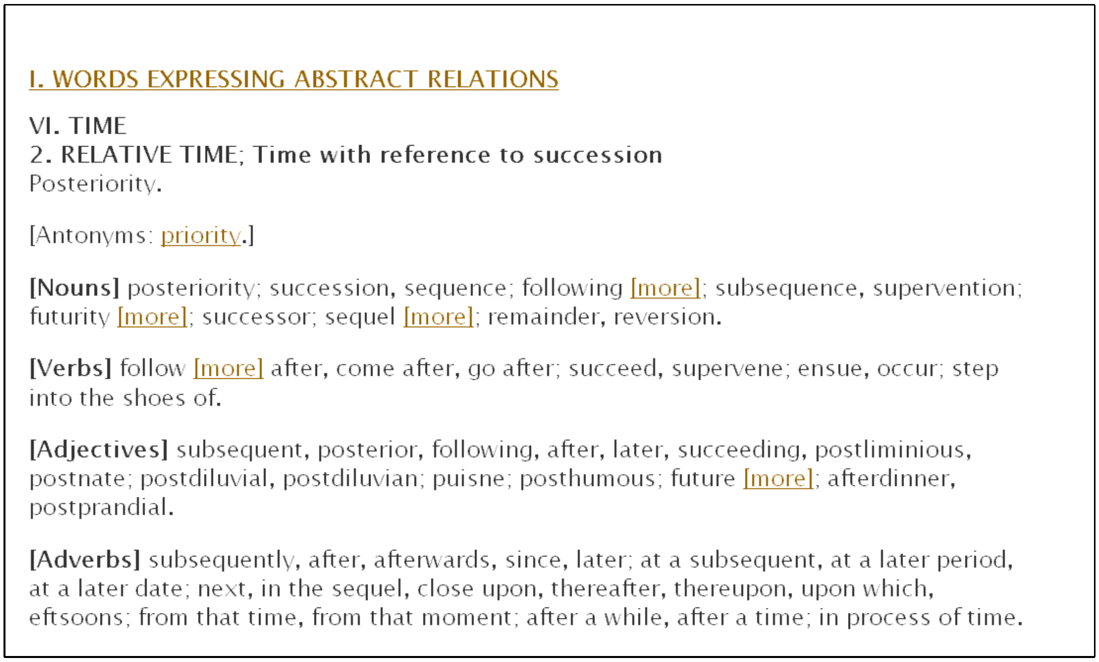
Thesauri

In a *thesaurus*, words are grouped by meaning and semantic distance. In a dictionary the entries are independent, whereas in a thesaurus all entries belong to at least one category and thus have relations to other entries. The idea of thesauri was invented by Peter Mark Roget and a first english thesaurus was published in 1852. Thesauri were originally intended as a memory-aid for writers in helping with finding the most appropriate word. The structure in *Roget's Thesaurus* from 1911 is to classify all words into approximately 1000 categories. In a thesaurus, a word may appear in more than one category due to words having multiple senses and the fact that there can be different perspectives on a single sense. Each thesaurus category is in turn divided into smaller groups of closely related words. Adjacent categories are often an indication of antonymous content. Table 4 contains category relation examples from Roget's thesaurus.

A thesaurus simply groups some related words, but does not specify the relationships. Thus, some of the words marked as related may not be contextually or semantically close. Thesaurus-based approach by Morris and Hirst (1991) is to define unnamed relationships based on structure of thesaurus, for example in-same-category-as and in-adjacent-category-to. Words are considered to be close, if they are in the same category or in categories that are related through index entries or cross-references. Figure 6 is a content example Roget's thesaurus with all hierarchical categories for word posteriority.

Related terms	***Relation description***
wife-married	both in same category
car-drive	category of car has cross-reference to category of drive
brutal-terrified	both are in categories with the same cross-reference to a third category

Table 4: Examples of thesaurus relations. (from Roget's thesaurus)



I. WORDS EXPRESSING ABSTRACT RELATIONS

VI. TIME
2. RELATIVE TIME; Time with reference to succession
Posteriority.

[Antonyms: priority.]

[Nouns] posteriority; succession, sequence; following [more]; subsequence, supervention; futurity [more]; successor; sequel [more]; remainder, reversion.

[Verbs] follow [more] after, come after, go after; succeed, supervene; ensue, occur; step into the shoes of.

[Adjectives] subsequent, posterior, following, after, later, succeeding, postliminious, postnate; postdiluvial, postdiluvian; puisne; posthumous; future [more]; afterdinner, postprandial.

[Adverbs] subsequently, after, afterwards, since, later; at a subsequent, at a later period, at a later date; next, in the sequel, close upon, thereafter, thereupon, upon which, eftsoons; from that time, from that moment; after a while, after a time; in process of time.

Figure 6: A content example for word "posteriority" in Roget's thesaurus.

The experimental part of this thesis contains code for using Roget's Thesaurus as one of the lexical resources in finding hierarchical relations between terms. Roget's Thesaurus has been implemented in Java as an Electronic Lexical Knowledge Base (Jarmasz and Szpakowicz, 2001). An 8-level hierarchy for grouping words and phrases in the thesaurus induces a measure of semantic distance between words and phrases (Jarmasz and Szpakowicz, 2003a). A distance is calculated as the length of the shortest path through the hierarchy between two given terms. A score reflects the level at which both words and phrases appear. Table 5 lists the fixed distances of the Jarmasz and Szpakowicz semantic distance calculation.

Semantic distance	***Term comparison condition based on Roget's categories***	***Example of category level***
0	the same Semicolon Group	succession, sequence;
2	the same Paragraph	(types of U.S. coins) penny, cent
4	the same Part of Speech	Nouns
6	the same Head	Posteriority
8	the same Head Group	Time with reference to succession
10	the same Sub-Section	2. RELATIVE TIME
12	the same Section	VI. TIME
14	the same Class	I. WORDS EXPRESSING ABSTRACT RELATIONS
16	different Classes or not found	penny, posteriority

Table 5: The Jarmasz and Szpakowicz semantic distance values and Roget's thesaurus categories. The Semicolon Group contains the most closely related terms, while the Class is the broadest category.

3.2.2 WordNet

The original *WordNet* is a large lexical database of English. It was developed under the direction of George A. Miller in Princeton University. The development started already in the late 1980's and versions of WordNet have since been developed for many other languages by many research groups (Miller, 1990; Fellbaum,1998).

In WordNet nouns, verbs, adjectives and adverbs are grouped into so-called *synsets*, sets of cognitive synonyms, each expressing a distinct concept. Synsets are the key building blocks of WordNet and they are interlinked by both conceptual-semantic and lexical relations. A word appears in one synset for each of its *senses*. Each synset has also a *gloss* (definition) and possibly some example use cases of the contained words.

The linking results in a navigable network of meaningfully related words and concepts. WordNet's structure makes it a useful tool for computational linguistics and NLP.

An example of WordNet structure:

```
dog, domestic dog, Canis familiaris
   => canine, canid
      => carnivore
        => placental, placental mammal, eutherian, ...
          => mammal
            => vertebrate, craniate
              => chordate
                => animal, animate being, beast, ...
                  => ...
```

In WordNet each synset is connected by the so-called IS-A relation to its *hypernyms* (more general concepts) and *hyponyms* (more detailed concepts), see Table 6. The IS-A relationship forms a set of trees in WordNet, in other words *a set of hierarchies* or taxonomies. The maximum depth of a synset is limited to 16 and multiple inheritance (same word belonging the multiple hierarchies) is allowed. WordNet can be used simply as a synonym hierarchy tree, but taking advantage of the provided additional relationships makes it a network. Examples of such relations are *antonymy* (opposite meaning), *meronymy* (is part of) and *holonymy* (has as part). A later developed project called EuroWordNet has many additional semantic relations and refinement of meronymy. The original idea behind EuroWordNet is to map meanings between different (european) languages.

The experimental part of this thesis uses WordNet as one of the lexical resources in finding hierarchical relations between terms.

3.2.3 FrameNet

FrameNet is an on-going project based on *semantic frames*, currently in its third release. A semantic frame can be thought of as a concept with a script. It is used to describe an object, state or event. The FrameNet lexical database contains around 10,000

Relation type	***Examples***
Synonymy or near-synonymy	mistake-error, command-order, enemy-foe (appear in same line in synset)
Subsumption hypernymy/hyponymy	apple-fruit, fruit-banana (appear in same synset hierarchy)
Meronymy and holonymy	engine-car, player-team, tree-forest, brick-house (PART-OF, HAS-A)
Antonymy	tall-short, big-small (COMPLEMENT-OF)

Table 6: Examples of WordNet relations.

lexical units (a pairing of a word with a meaning; polysemous words are represented by several lexical units), 800 semantic frames and over 120,000 example sentences (Fillmore et. al, 2010). FrameNet is a project similar to WordNet. It consists of a lexicon which is based on human annotation of over 100,000 sentences with their semantic properties. The unit in focus is the semantic frame, a type of state or event together with the properties associated with it.

Note that FrameNet is not an *ontology*, a complete conceptual model of the world, instead it is an attempt to create semantic frames. Such frames contain a limited set of other concepts related to the concept at hand. Semantic frames make it possible to make general statements about the semantic-syntactic behavior of *groups of lexical units*, rather than one at a time. FrameNet contains more than 1000 frames and the developers have defined, with computer-aided annotation, a rich structure of relations between them (Table 7). The relation mappings partially form an inheritance hierarchy. The upper frames in that hierarchy resemble some of the upper nodes in existing ontologies. The goal of FrameNet, however, is to fully represent the linguistic facts, rather than to exhaustively categorize the entities and events in the world. FrameNet developers have created a set of semantic types which are applied to frames, frame elements, and lexical units.

A *lexical unit* (LU) is used in pairing a word with a meaning. Words can have multiple senses i.e. meanings in different contexts. Each sense of a polysemous word belongs to a different semantic frame, a script-like conceptual structure that describes a particular type of situation, object, or event along with its participants and props. For example, in the Apply_heat frame (Table 8), the Cook *frame element* (FE) has the semantic type

A *Cook* applies heat to *Food*, where the *Temperature_setting* of the heat and *Duration* of application may be specified. A *Heating_instrument*, generally indicated by a locative phrase, may also be expressed. Some cooking methods involve the use of a *Medium* (e.g. milk or water) by which heat is transferred to the *Food*. A less semantically prominent *Food* or *Cook* is marked *Co_participant*.

Table 7: Description data from a FrameNet frame Apply_heat.

"Sentient", the Container, the semantic type "Container" and the Heating_instrument, "Physical_entity". Some of the categories are rather broad, but the idea is that they can help a semantic parser pick out the right pieces of a sentence to label with these FEs.

Core frame element	***Semantic type***
Container	Container
Cook	Sentient
Food	
Heating_instrument	Physical_entity
Temperature_setting	Temperature

Table 8: Core frame elements of frame Apply_heat.

In addition to the core frame elements, *non-core frame elements* have been defined for the frame Apply_heat. In the example, in Table 9, the FEs Medium and Co_participant are non-core FEs. All others are semantically essential, core parts related to applying heat.

Cook	***Apply_heat***	***Food***	***Medium***	***Heating_instrument***	***Co_participant***
Sally	fried	an egg	in butter		
Sally	fried	an egg		in a teflon pan	
Ellen	fried	the eggs			with chopped garlic

Table 9: Examples of frame elements related to Apply_heat.

The FrameNet frames and thus the Frame Elements and Lexical Units associated with them, are intended to be situated in semantic space by means of frame-to-frame relations and semantic types. The relations used in FrameNet include Inheritance, Subframe, Causative_of , Inchoative_of , and Using. Table 10 presents an example of frame to frame relations from the Apply_heat frame.

Relation type	***Relations of frame Apply_heat***
Inherits from:	Activity, Intentionally_affect
Is Used by:	Cooking_creation
Is Causative of:	Absorb_heat

Table 10: Frame-frame Relations for frame Apply_heat.

From the sentence topic modeling point of view, WordNet provides lexical hierarchical relations and FrameNet provides semantic hierarchical relations. The FrameNet relation types in table 11 include for example causative and temporal relations. The relations in FrameNet exist between frames (collections of related concepts), whereas the WordNet relations exist between synsets (near synonymous words).

Relation	***Sub***	***Super***
Inheritance	Child	Parent
Perspective on	Perspectivized	Neutral
Subframe	Component	Complex
Precedes	Later	Earlier
Inchoative of	Inchoative	State
Causative of	Causative	Inchoative/State
Using	Child	Parent
See also	Referring Entry	Main Entry

Table 11: Types of Frame-frame relations, from (Fillmore et. al, 2010).

3.2.4 Ontologies

There are many existing ontologies and knowledge bases which aim at modeling the general concepts of the world around us. These are called upper ontologies or world ontologies. A more domain or context oriented approach is the use of frames in mapping object properties and contextual co-occurrence. This contextual approach was discussed in more detail in Section 3.2.3. The existence of broad coverage public resources, like Wikipedia and Wiktionary, has encouraged researchers to attempt semi-automatically constructing such ontologies.

One of the problems with ontologies is that an ontological entity is a *surrogate* for the thing itself in the real world. A surrogate is a description, often just a name perhaps enriched with a few properties, instead of the “real entity” in the real world or the conceptual model of the real entity in your head. This is due to the fact that defining the “real entity” is difficult.

> *The word “bicycle” is a surrogate for a transportation device. But what does the surrogate describe? My bicycle, an instance of bicycle, the concept of bicycle, ...*

How accurate can a surrogate become, if we add properties to it? In theory it is possible to add all meaningful properties to a surrogate, but in practice using as high fidelity as possible in the description is problematic.

> *My “bicycle” is green and has received good maintenance. But how did I perform the maintenance? Should I mention that I opened and lubricated all moving parts? Which moving parts, all bolts and screws too?*

Comprehensively describing a concept with all its properties, relations and contextual connections is next to impossible. This is because a general description of a concept is always an approximation and a compromise. An approximation, because no common description exists which would fit all situations where a bicycle needs to be mentioned. A compromise, because you cannot or need not tell everything related every time you talk about a bicycle. Thus you can describe a bicycle based on your personal conceptual model, but the reader will interpret your message with his own conceptual model.

The experimental part of this thesis does not extend to artificial intelligence, which would mean attempting to interpret all ontological relations. Instead the focus, in use of external resources, is on finding simple hierarchical relations and other words from the same context.

4 Relevance and semantic relatedness measures

Relevance is defined as how connected or applicable something is to a given matter. Philosophically, a thing is relevant if it serves as a means to a given purpose, in other words, a sentence is relevant to a domain when it's content describes or discusses the underlying whole text context. Sentence relevance can be indicated, for example, by collecting a domain vocabulary and matching sentence words to the vocabulary.

In the experimental part of this thesis, relevance measuring is used in finding key sentences of a text. In a text, many sentences contain similar or related content and thus a semantic relatedness measure is used in ranking the sentences with related content.

4.1 Domain vocabularies as relevance measures

A *domain vocabulary* is the set of words within a language that are specific to a given domain, for example, the terminology used in computer science. In human language learning, a vocabulary usually grows and evolves with time, and serves as a useful and fundamental tool for communication and acquiring knowledge. Acquiring an extensive domain vocabulary is one of the difficulties in using a vocabulary as a relevance measure.

Extensive domain vocabularies can be collected by finding keywords from large text corpora. This would result in a reliable vocabulary, but collecting one is a time-consuming task. In the experimental part of this thesis, a more economical approach of collecting domain vocabulary from a text is studied. Rather than finding the most common keywords from corpuses of given domain, we adopt the opposite approach. That is, we detect domain keywords by their *lack of presence* in a 150 000 common word lexical resource, in this case WordNet (Fellbaum, 1998). Furthermore, words co-occurring in the same sentence with domain words are given half the relevance value of a domain vocabulary word, all others are treated as common words.

The pros of using a domain vocabulary as a relevance measure include that it works well with scientific texts which contain a lot of special terminology. The cons include that the approach does not work for general descriptive texts and has a heavy dependence on the used lexical resource content. The same problem, dependence on training data, is present also when directly collecting keywords from a seemingly large corpora,

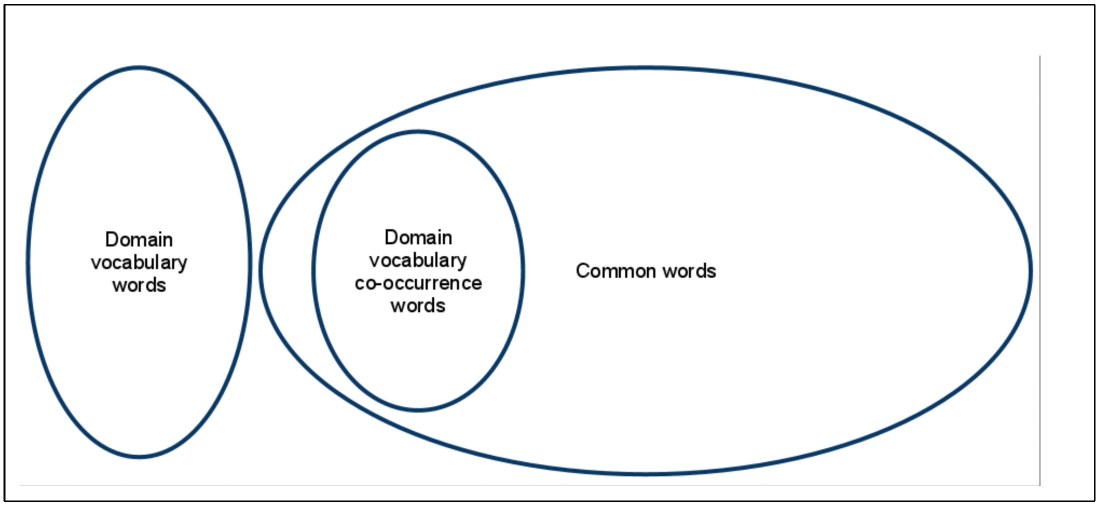


Figure 7: Domain vocabulary words are those not found in common word resources like WordNet. Words co-occurring in same sentence with a domain vocabulary word are marked as domain vocabulary co-occurrence words. Domain vocabulary words are given a relevance value (1), co-occurring words half the relevance value (0,5) and common words are treated as irrelevant (0). Each occurrence of a domain vocabulary word or co-occurring common word adds to the total relevance of a sentence.

which still does not contain all central terms of a domain.

A domain-specific *ontology* would be better than a domain vocabulary. Unfortunately, such ontologies have not been assembled for most domains and constructing one automatically is a non-trivial task. A domain ontology models a specific domain, or part of the world and represents the particular meanings of terms as they apply to that domain. An ontology would include a gloss for each word and would separate the multiple senses of one word and have hierarchical relations between the words.

4.2 Semantic similarity vs. semantic distance

Semantic similarity means detecting synonymous words, whereas *semantic distance* is used for constructing a numerical distance between words of different meaning. One approach to both problems is to first look for known related words from an external source (or just words appearing nearby in text) and constructing relation chains or graphs between found words. This is much like constructing fractions of a mind map of the general topic.

In cohesive and coherent texts, which is expected of scientific texts, sentences are likely to refer back to previously introduced concepts. In texts written to readers already familiar with the general topic, references are also made to related concepts that have not been introduced in the written text. All these references form *lexical chains* in the text (Hirst and Budanitsky, 2001).

Formal relations

There are two major categories of lexical semantic relations. *Formal relations* like IS-A and HAS-A have been collected to lexical resources (WordNet for example). *Typicality relations*, in turn, are bindings to the usage context of words, most often binding other sentence words to the action (verb). Typicality is not a generic feature across all texts, but rather domain and situation specific. Examples of formal relations are similar meanings in synonymy, hierarchical relations like hypernymy and hyponymy and opposite meanings in antonomy.

```
chair IS-A seat IS-A furniture IS-A ... IS-A thing

house HAS-A porch
house HAS-A loft
```

Figure 8: An example of IS-A and HAS-A formal relations. IS-A relations form a hierarchical chain of higher and higher level concepts (chair –> thing), whereas HAS-A relation is pair of higher level concept (house) and a lower level, contained concept (porch).

Typicality relations

Examples of typicality relations are co-occurrences of agent (subject) and patient (object) in an action or event (verb). Detecting causality is mainly done by detecting cue words like "because" and then collecting the agent, action and patient from the sentence to a typical causal relation. Purpose of use can also be detected with cue words such as "with". Simply counting word co-occurrences also provides information about typical relatedness. Examples of typical co-occurrences are snow and ice, bread and butter and such.

Lexical chains and semantic distance

```
Detecting/V cue words helps in finding/V
causality relations.

Finding/V purpose of use is quite similar, it
can be detected/V with cue words like "with".
```

Figure 9: An example of typicality relations, specifically the term “cue words” co-occurring with verbs “find” and “detect”.

Lexical chains are lists of related words in a text (Halliday and Hasan, 1976). Words are added to an existing lexical chain only if it is related to any of the words in the chain by a cohesive relation. Two main approaches to approximating cohesion are term reiteration and semantic distance calculation. An example of term reiteration approach is given in Figure 10. The example sentences used in the table are S1="Boundaries are set between sequential segments of text", S2="Text is often subdivided into paragraphs", S3="more to achieve visual layout that aids reading" and S4="than to indicate a change in the topic under discussion".

Each word in a lexical chain is related to its predecessors either by a) identity of reference or by b) being somehow semantically close or semantically related (Hoey, 1991). For the first category, *reference resolution* is the subtask and forms its own research field in NLP. The second category, semantic relatedness, is a broad term. It includes for example the NLP subtasks of SRL and WSD.

4.3 Methods for computing semantic relatedness

Semantic relatedness is typically computed using one of the two main classes of methods. *Resource-based measures* rely on an external linguistic resource in calculating semantic distance, where as *distributional similarity measures* depend on co-occurrence data. Co-occurrence data can be collected from the document at hand, but a higher quality resource; one with more terms and relations, can be constructed from a large text corpora.

Resource-based measures

Sentence	***Chain from sentence***	***Active Chains after***
S1	boundary-segment-text	boundary-segment-text
S2	text-paragraph	boundary-segment-text-paragraph
S3	layout	boundary-segment-text-paragraph, layout
S4	change-topic-discussion	layout, change-topic-discussion

Figure 10: Simple lexical chain construction with term reiteration (nouns). Chains are constructed by checking if already encountered words reappear in the following sentences. When no words of a chain are found (in a given window of, for example, three sentences), the chain is closed. New chains are started whenever a word is not found in the still active chains.

Morris and Hirst (1991) used thesaural relations in constructing lexical chains and Jarmasz and Szpakowicz (2003a, 2003b) have used Roget's thesaurus in detecting semantic similarity. (Refer to Section 3.2.1 for details of thesaural approaches.) In text summarization, Barzilay and Elhadad (1997) used relations between WordNet (Fellbaum, 1998) synonym-sets to estimate semantic distance. Later Barzilay and Elhadad (1999) have shown that a sentence or paragraph with many lexical chains running through is likely to be a good choice to include in a summary.

Resource-based methods, using a dictionary or thesaurus for example, are an improvement over term reiteration and they capture much larger amount of semantic information. However, their dependence on a specific resource is problematic. The methods are often unable to operate across parts of speech (POS) or consider other than class relations. Lemmatization (see Section 3.1.3) can ease comparisons between words of different POS. Use of dictionaries or ontologies, which contain more descriptive information, can help in finding other than class relations.

Measures of distributional similarity

Measures of distributional similarity rely on word co-occurrence information to calculate semantic distance. Unlike resource-based measures, these measures are not affected by the limitations of a specific linguistic resource. For example, WordNet contains 150 000 common words but very few words specific to scientific domains. The downside of distributional measures is that they often run into word sense ambiguity problems, because they consider only the surface forms of words and not the word senses (meanings). Also, their correlation with human judgments is observed to be fairly low (Weeds, 2003).

Hybrid methods combining resource-based and distributional measures

The shortcomings of both resource-based measures and distributional measures, when used alone, rises the need for a method that incorporates the advantages of both. Mohammad and Hirst (2006) proposed measures for combining distributional co-occurrence information with semantic information from a thesaurus. These measures were shown to outperform traditional distributional measures on the tasks of correcting real-word spelling errors and ranking word pairs in order of semantic distance (Mohammad and Hirst, 2006).

The framework of Mohammad and Hirst, which they dubbed as *distributional measures of concept distance* (DMCDs), combines distributional word co-occurrence information with the semantic information from a thesaurus. A DMCD is configured by choosing an appropriate *window size* (they used 5 words before and after), the measure of distributional similarity, and the statistic used to measure the strength of association. DMCDs were evaluated by ranking word pairs in order of their semantic distance with human norms. DMCDs outperformed all distributional measures on the task, but stayed second to the Jiang and Conrath (1997) WordNet-based measure.

Computing semantic relatedness

How to compute *semantic closeness*? A human reader instinctively has an intuition about the semantic distance between two words. A computer, however, does not have a world model against which to reflect it's observations.

Construction of lexical chains depends on semantic closeness, but how can we determine the semantic distance between two words and whether that distance is small? Converting a human intuition about semantic distance to a computational algorithm is still an open research problem. Some attempts on structuring the problem have been made in the past. Rubenstein and Goodenough (1965) and later Miller and Charles (1991) constructed an experiment where they asked people to judge semantic distance between given pairs of words on a scale from 0 to 4. Examples from the test categorizations are a) highly synonymous: gem-jewel, b) semantically related: crane-implement and c) semantically unrelated: noon-string. The answers from test subjects were consistent (90 percent correlation) and the researchers concluded that there are different types (Table 12) of semantic relations between the words.

Word pair	***Relatedness***
car,automobile	3.92
gem,jewel	3.84
...	
crane,implement	1.68
journey,car	1.16
...	
rooster,voyage	0.08
noon,string	0.08

Figure 11: A sample of the results from Miller-Charles semantic distance judgment experiment. The relatedness values are on a scale from 0 to 4 and a high value indicates close relatedness. Human test subjects considered “car” and “automobile” to be closely related and “noon” and “string” to have next to nothing in common.

Human intuition about semantic distance

Word meaning and semantic distance between words can be approximated with the help of dictionaries and other lexical resources. However, all world knowledge is not available for a computer, although ontologies are being constructed for the purpose. Context frames are a simpler approach to modeling properties and occurrences in a similar context. The psychology category is challenging to compute, though its use is very typical and effortless in human everyday thinking. An average european can associate Italy with red sports cars or with pasta, even though the terms do not have similar definitions or properties. A key problem is modeling the world in a way that guides traversing the semantic relatedness and context model. Table 12 lists the categories and intuitions behind them.

Category	***Intuition***
Word meaning	definitions are related or similar in some way (vehicle, bus)
World knowledge	two things have similar properties or often occur together or in a similar context (car, driving)
Psychology	we often think of the two things together (apple, banana)

Table 12: Human intuition about semantic distance.

Guidance for traversing the model could be implemented in a model which consists of context-bound, weighted relations so that only the most essential relations of a concept can be retrieved. The idea is the same as in calculating information content (IC), the occurrence frequency, for words. In this case the frequency would be calculated for word pair relations or n-gram relations.

WordNet::Similarity library for computing semantic relatedness

WordNet::Similarity (Pedersen et. al, 2004) is a Perl module that implements a variety of semantic similarity and relatedness measures based on information found in the lexical database WordNet. In particular, it supports the measures of Resnik, Lin, Jiang-Conrath, Leacock-Chodorow, Hirst-St.Onge, Wu-Palmer, Banerjee-Pedersen, and Patwardhan-Pedersen. In the experimental part of this thesis, a Java implementation of the same measures is used (from David Hope, University of Sussex, http://www.cogs.susx.ac.uk/users/drh21/).

There are three main approaches to semantic relatedness measuring. The measures in WordNet::Similarity can be categorized to path, information content and gloss overlap measures. Currently the Perl version also contains a lexical chain construction algorithm.

For semantic relatedness measuring in the experimental part of this thesis, Jiang-Conrath measure was selected. This measure has produced the best results in many comparisons (Jurafsky and Martin, 2009) and it has been shown to outperform a combination of distributional and resource-based measures (Mohammad and Hirst, 2006). The algorithm uses WordNet relations and pre-calculated *information content* (IC) values in selecting the *least common subsumer* (LCS). See Figure 13 for a visualization of IC and LCS.

Information content (IC) is the word occurrence frequency in a text corpora. Function IC(x) is used in expressing the information content of x.

Least Common Subsumer (LCS) is the most informative subsumer (the most frequently occurring word) in the path between the two words.

Figure 12: Definitions of information content and least common subsumer.

The creators of WordNet::Similarity have pre-computed information content files from the British National Corpus, the Penn Treebank, the Brown Corpus, the complete works of Shakespeare and SemCor. A Perl script is available from the WordNet::Similarity website, with which the IC files were created (http://search.cpan.org/ tpederse/WordNet-Similarity/utils/rawtextFreq.pl).

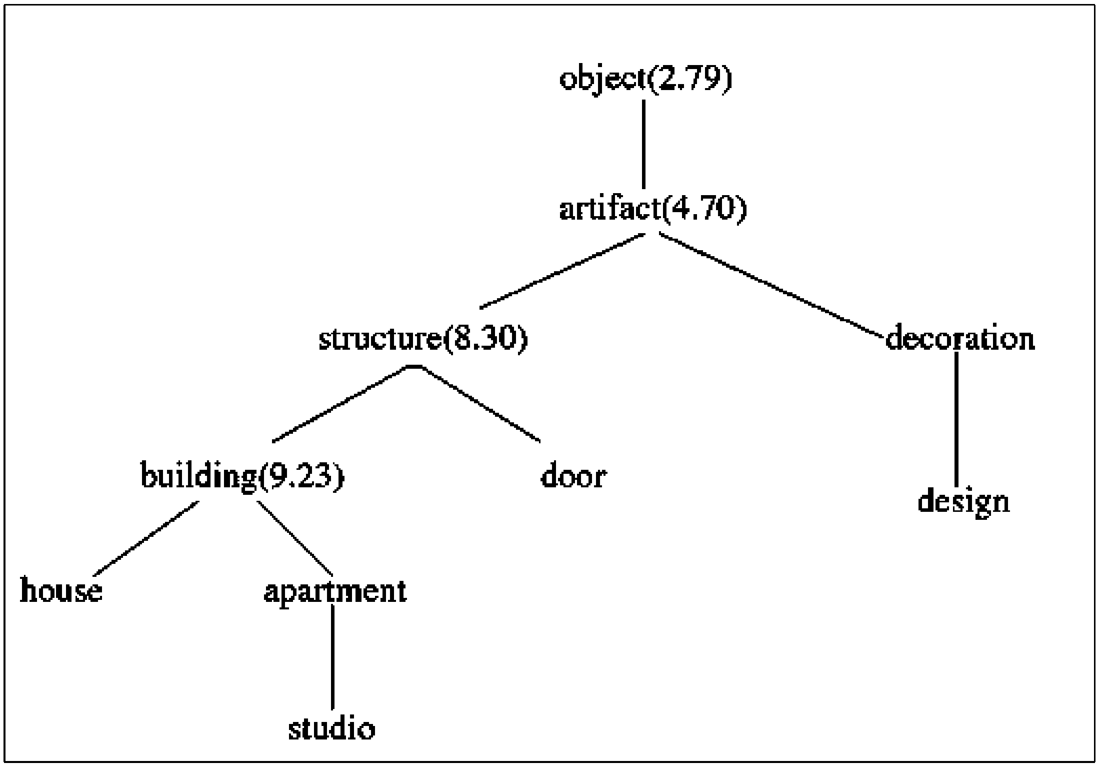


Figure 13: Example of information content in WordNet. When calculating the relatedness between “house” and “decoration”, the least common subsumer will be the word “building”, because it is on the path between the words and has the highest information content value of IC=9.23. We only included the information content values on the path and in the shared higher level concept in the figure, in order to point out which IC values are checked when determining the LCS of two words.

The relatedness value returned by the Jiang-Conrath measure is equal to $1/jcn\_distance$, where jcn_distance is equal to:

$$IC(synset1) + IC(synset2) - 2 * IC(lcs)$$

In the equation synset1 is the WordNet synonym set identifier of the first term (house) and synset2 is the WordNet synonym set identifier of the second term (decoration).

5 Proposed relevance modeling algorithm

We propose an algorithm for finding essential, domain-specific content from scientific texts. The motivation in constructing such an algorithm is in the readers' need of quickly finding the core content, at best the contribution, of a scientific text. A similar need exists in many other fields, for example, in natural language generation and document indexing. In the experimental part of this thesis, the proposed algorithm is applied to extractive text summarization. The algorithm first pre-processes the text with NLP tools. This includes splitting the text to sentences, tagging each word with a part-of-speech and lemmatizing each word to it's dictionary form. Secondly, a domain vocabulary is collected and relevance values calculated for all sentences based their relation to the collected vocabulary. In the third step, relatedness values between all sentence pairs are calculated with a similarity measure that uses WordNet.

After all features have been extracted, the relevance of sentences is determined by categorization. From the categorization results an extractive summary of the relevant text content is constructed. The following list summarizes the algorithm steps.

1. *Split text to sentences:* Construct the segments for which topic signatures will be generated.

2. *Pre-process with NLP tools:* Extract words from sentences, use POS and lemmatization to improve comparability of words.

3. *Collect domain vocabulary:* Construct domain vocabulary from words not present in WordNet, which contains only common words.

4. *Calculate domain specificity:* Use the collected domain vocabulary and co-occurring common words to evaluate sentence relevance to the whole text.

5. *Calculate semantic relatedness:* Use *WordNet::Similarity* library to calculate semantic relatedness values between sentences. This is done in order to find sentences that are (semantically) most connected to other parts of the text.

6. *Categorize sentences:* Normalize the three calculated values. Categorize sentences to relevant and irrelevant using sentence data rows with the calculated domainVocabulary, coOccurrence and semanticDistance values.

5.1 Constructing topic signatures

There are three steps in constructing topic signatures: 1) extracting features, 2) filtering the features and 3) enriching the features. The steps indicate that we can extract useful features, like individual words, from the text. But some features are too common to be used in categorizing a sentence which contains them. On the other hand, the source text seldom contains all words needed to express the underlying semantic content, the meaning and broader context of the text.

Extracting features

In the extraction step, text is split to segments; to sentences, phrases and words. This may seem like an obvious and simple step, but it is non-trivial for a computer, which does not have eyes, brains or world knowledge. Current state-of-the-art text segmentation tools have statistical models for estimating the sentence boundaries. (Refer to Section 2.1 for discussion on text segmentation.)

Natural languages have a structure and making the structure available to an algorithm requires the use of NLP tools. Finding the part-of-speech of each word, for example whether the word is a noun or verb, is explained in more detail in Section 3.1.1. Lemmatization means finding the dictionary form of a word. This is done to improve comparability of words and is applicable mostly to words belonging to the same part-of-speech. Lemmatization and the difference to stemming with strict string matching is discussed in Section 3.1.3.

Filtering features

All extractable features of a sentence are not essential in connecting the sentence to a broader context. One definition of *essential* is the ability to make distinctions between candidate concepts. This means that more domain-specific and rarely occurring concepts make a sentence more unique.

The problem for an algorithm is finding essential content and modeling only that. The approach taken in this thesis includes collecting a domain vocabulary, calculating word occurrence frequency and leaving out too common words with a stop word list. Stop word lists and other semantic smoothing methods are discussed in Section 2.3.

Some of the typically used, and proven to be distinctive (Lin and Hovy, 2000), features in text summarization are position, cue phrases, word informativeness, sentence length and cohesion. In news feed texts, first sentences of paragraphs are often included in summaries. Also sentences containing highly domain-specific words indicate extract-worthiness, whereas too short sentences seldom include enough information to be included in a summary. Content selection and cues are discussed in more detail in Section 2.2.

Enriching features

One of the key limitations of local features, those extractable from a text without external resources, is that all the terms needed in understanding and relating the sentence to a broader context are not included in the text. But even the external resources have their limitations. Even the broad coverage, human-constructed resources like WordNet (Fellbaum, 1998) do not include all terms and relations. Even these state-of-the-art resources often have little or no binding to the occurrence contexts of words.

The limitation of using external resources in enriching sentence features is that each measure is only as good as the resource it depends on. Most WordNet measures, included in WordNet::Similarity library, use only nouns and IS-A relations, algorithm parameters are set with very sparse human data and role of context in semantic distance judgments is not accounted for. To tackle these limitations Hearst (1994) suggests use of statistical word co-occurrence data.

The enriching approach used in this thesis is to use only immediate context, the closest related hypernyms of sentence words. Direct hypernyms for sentence subject, verb and object lemmas are queried from WordNet. This approach could be easily extended by including multiple levels of hypernyms and hyponyms. The steps of enriching topic signatures were explained in detail in Section 2.3.

5.2 Computing domain specificity

Domain specificity of a sentence is approximated by calculating two values, the domain vocabulary word occurrence count and the occurrence count of words co-occurring with domain vocabulary words in other sentences. For example, one sentence mentions “flying a spaceship” and another sentence mentions “flying”. In the first sentence “spaceship” is domain-specific and it causes “flying” to be treated as a co-occurrence word.

The following pseudo code summarizes the algorithm steps, the actual code produced in the experimental part of this thesis is written in Java and is available as part of our Scientific Writing Assistant project (SWAN, http://cs.uef.fi/swan/) source code.

```
// collect domain vocabulary
for each sentence in document
  for each word in sentence
    if (WordNet does not contain word)
      addToDomainVocabulary(word);

// collect domain vocabulary word co-occurring common words
for each sentence in document
  for each domainVocabularyWord in domainVocabulary
    if (sentence contains domainVocabularyWord)
      for each commonWord in sentence
        addToCoOccurrenceVocabulary(commonWord);

// calculate domain specificity values of sentence
for each sentence in document
  int domainSpecificity = 0;
  int domainWordCoOccurrence = 0;

  for each word in sentence
  {
    if (domainVocabulary contains word)
      domainSpecificity++;

    if (domainVocabularyCoOccurrenceWords contains word)
      domainWordCoOccurrence++;
  }
```

Collecting domain specificity values is done as part of collecting sentence categorization input data. Refer to Table 13 for examples of normalized input data to sentence categorization. The following graphs show calculated domainVocabulary and coOccurrence values.

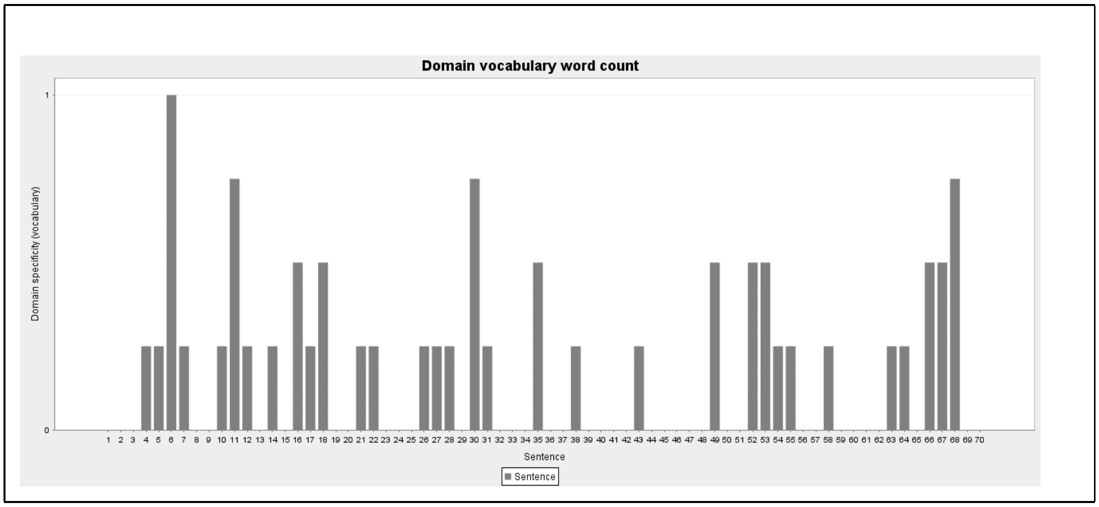


Figure 14: Domain specificity of sentences from first test document according to domainVocabulary parameter. The 70 sentences of first test document are shown on the x-axis and the domainVocabulary value (normalized to 0..1) on the y-axis. The same representation is used also in the following coOccurrence and semanticDistance parameter bar charts.

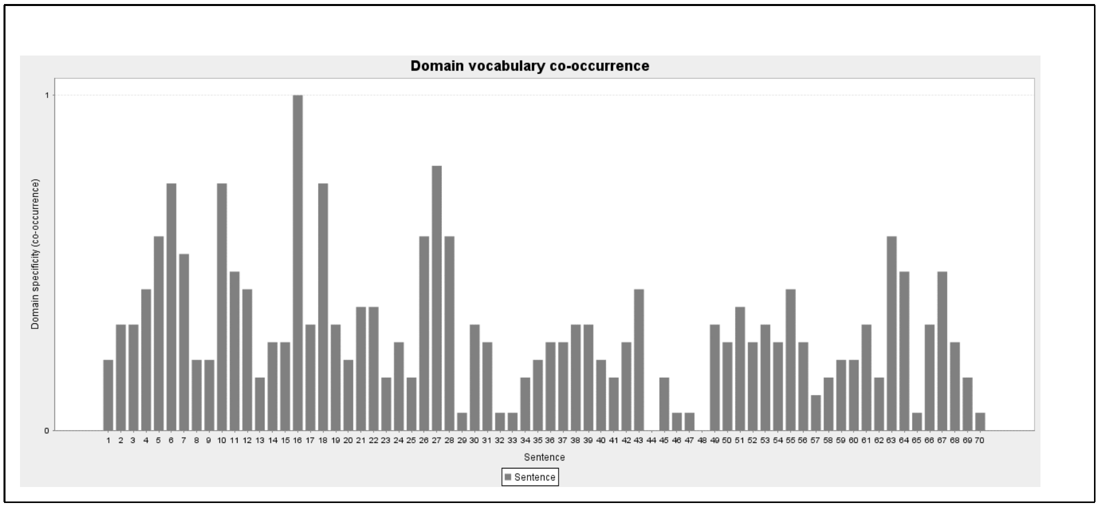


Figure 15: Domain specificity of sentences from first test document according to coOccurrence parameter.

5.3 Computing semantic relatedness

Semantic relatedness of a sentence, to all other sentences in the same document, is approximated by calculating one average relatedness value. This is achieved by first computing relatedness between sentence words, then constructing a temporary sentence pair relatedness matrix and finally calculating average relatedness value from each matrix row. Both row and column averages are calculated in order to reduce the effect of varying sentence length.

```
// calculate sentence term matrix row averages
for each row
    rowAverages[i] = rowTermsAvgSum / rowCount;

// calculate sentence term matrix column averages
for each column
    colAverages[i] = colTermsAvgSum / colCount;

rowAveragesSum = sum(rowAverages);
colAveragesSum = sum(colAverages);

avgRelatedness = (rowAveragesSum / rowCount * colCount)
               + (colAveragesSum / colCount * rowCount);
```

The following pseudo code summarizes the algorithm steps, the actual code produced in the experimental part of this thesis is written in Java and is available as part of our Scientific Writing Assistant project (SWAN, http://cs.uef.fi/swan/) source code. Jiang and Conrath semantic relatedness measure is discussed in more detail in Section 4.3.

```
int[][] semanticDistanceValues;
int[] avgSemanticRelatednessValues;

// calculate semantic relatedness between all sentence pairs
for each sentence in document
  for each otherSentence in document
  {
    int semanticDistance = 0;

    semanticDistanceValues[sentence][otherSentence] =
      calculateJiangConrathRelatedness(sentence, otherSentence);
  }

// calculate average relatedness values
for each sentence in document
  avgSemanticRelatednessValues[sentence] =
    calculateAvgRelatedness(semanticDistanceValues[sentence]);
```

Collecting semantic relatedness values is done as part of collecting sentence categorization input data. The following graph shows calculated average semanticDistance values. Average is calculated from relatedness to all other sentences of the same document.

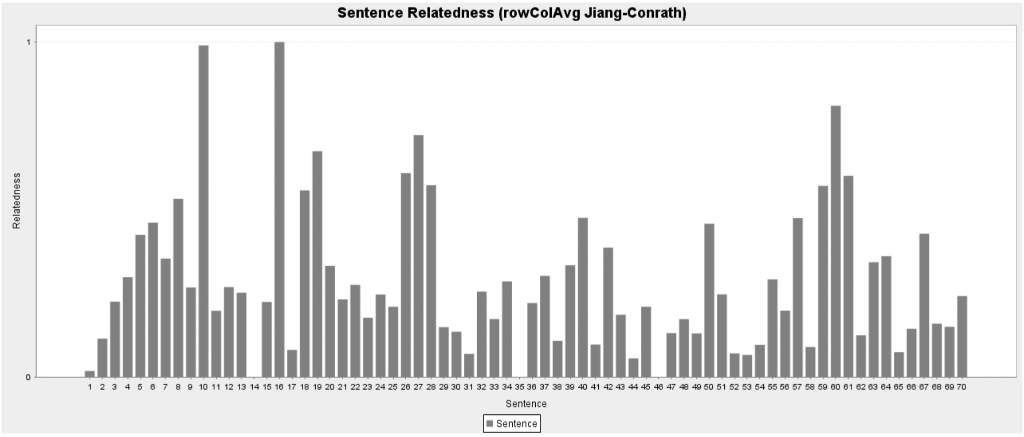


Figure 16: Sentence semantic relatedness of first test document according to semanticDistance parameter.

5.4 Categorization to relevant and irrelevant

Common approaches to providing sentence content for categorization include: 1) as such without pre-processing, 2) as stemmed word vectors or 3) as *n-grams* of sequential word occurrences. We instead experiment by first pre-processing the sentences with Stanford NLP tools, then constructing a topic signature of three latent sentence features for each sentence.

The sentence latent features are first calculated in a pre-processing phase and data rows with these values are then categorized. The domainVocabulary value is a sum of domain vocabulary words in the sentence, coOccurrence is a sum of common words occurring together with any of the domain vocabulary words. The semanticDistance value represents the semantic connectedness of this sentence to other sentences of the text. A simple arithmetic average value for each of these attributes, encountered in a test document, is calculated. For domainVocabulary and coOccurrence an above average value, and for semanticDistance a below average value, indicates relevance. A sentence is categorized as relevant if more than one attribute indicates relevance.

Sentence	***domainVocabulary***	***coOccurrence***	***semanticDistance***
S1	0	0.315789	0.114542
S2	0	0.315789	0.224857
S3	0.25	0.421053	0.29816
S4	0.25	0.578947	0.424506
S5	1	0.736842	0.460848

Table 13: Example of input data to sentence categorization. This data can be stored in the topic signature of a sentence and thus no external resources or matrices are needed in the categorization phase.

Histograms of input data distribution divided into 10 bins. Figure 17 shows domainVocabulary parameter values distribution, Figure 18 coOccurrence parameter and Figure 19 semanticDistance parameter.

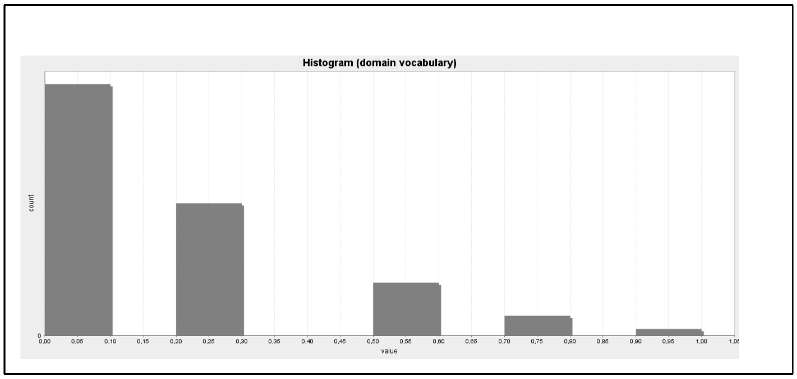


Figure 17: Histogram of domainVocabulary parameter from first test document.

The histograms show that most sentences have a low attribute value. Considering domainVocabulary and coOccurrence attributes, his means that only few sentences are domain specific. The large number of sentences with short distances to others makes categorization harder using only semanticDistance attribute. In the case study, all three

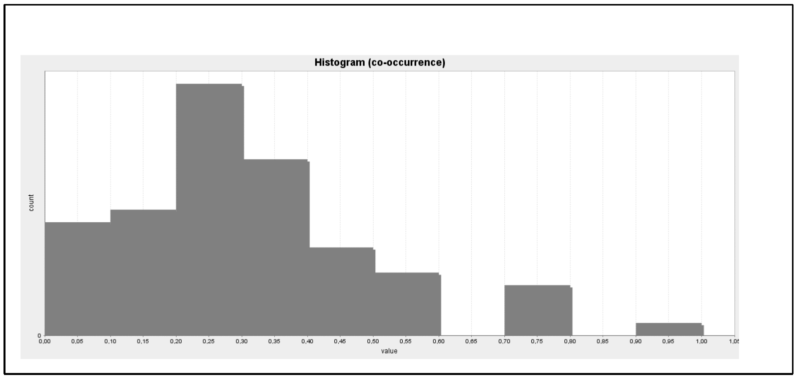


Figure 18: Histogram of coOccurrence parameter from first test document.

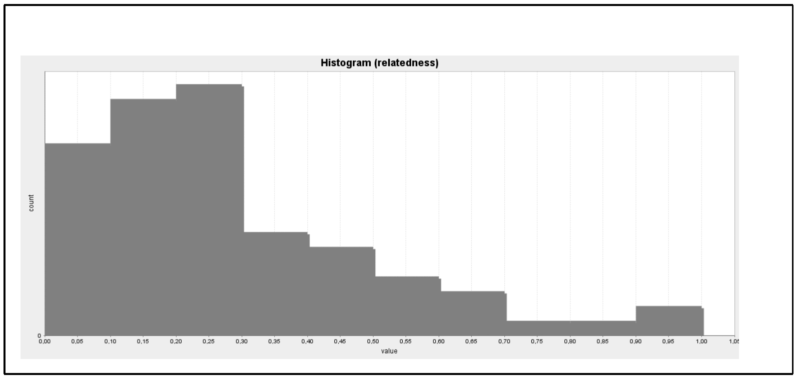


Figure 19: Histogram of semanticDistance parameter from first test document.

attributes are considered when categorizing each sentence.

6 Case study: Extractive summary generation

In this chapter we implement and evaluate the method of Section 5 in the context of extractive text summarization. Text summarization means compressing the source text into a shorter version while preserving its information content and overall meaning.

6.1 Methods and input data

The following four published scientific documents were used as input data for the summarization algorithm. The title, abstract, introduction and conclusions sections of the documents were first collected to plain text files (testDoc1, testDoc2, testDoc3 and testDoc4). According to scientific writing expert Jean-Luc LeBrun (www.scientific-writing.com), these are the key sections of a scientific paper, when it comes to providing an overview of the whole paper to the reader.

- *testDoc1*: V. Hautamäki, T. Kinnunen and P. Fränti, “Text-Independent Speaker Recognition Using Graph Matching”, Pattern Recognition Letters, 29(9): 1427–1432, 2008

- *testDoc2*: R. Saeidi, J. Pohjalainen, T. Kinnunen, P. Alku, “Temporally Weighted Linear Prediction Features for Speaker Verification in Additive Noise”, Odyssey 2010: The Speaker and Language Recognition Workshop, Brno, Czech Republic, pp. 40-46, June 2010.

- *testDoc3*: J. Sandberg, M. Hansson-Sandsten, T. Kinnunen, R. Saeidi, P. Flandrin, P. Borgnat, “Multitaper Estimation of Frequency-Warped Cepstra with Application to Speaker Verification”, IEEE Signal Processing Letters, 17(4): 343–346, April 2010.

- *testDoc4*: T. Kinnunen and H. Li, “An Overview of Text-Independent Speaker Recognition: from Features to Supervectors”, Speech Communication 52(1): 12–40, January 2010

The following list summarizes the steps of applying the proposed algorithm to extractive text summarization:

1. *Collect relevant sentences:* Collect relevant sentences with the proposed algorithm.
2. *Construct extractive summary :* Construct an extractive summary of the source text from the collected relevant sentences. Summary construction steps include 1) selecting number of sentences to include in summary, 2) selecting sentences from categorization results, in our case study, maximum number of ten sentences, 3) ordering selected sentences. Simple approach to ordering is to use order of occurrence in the source text. When the suggested context words improvement is available, ordering can be done by grouping sentences to latent topics with context word matching.
3. *Compare summary to abstract:* Compare the constructed summary to the human written abstract of the same scientific text by calculating term overlap.

Evaluation of text summarization systems

Text summarization systems and natural language generation systems are evaluated by their solutions to three key problems: 1) content selection, 2) information ordering and 3) sentence realization (Jurafsky and Martin, 2009).

In this case study, content selection is done by calculating three latent values for each sentence and then categorizing the sentences, as described in Section 5. First a set of relevant sentences is calculated, which aims at reducing the number of summary candidate sentences. Then information content of sentence (sum of contained words ICs) is used in selecting ten sentences to include in the extractive summary. The information content values used are the semcor values from Wordnet::Similarity.

Information ordering is kept unchanged; it is the same as the order of appearance in the source text. A future improvement would be to use the sentence semantic relatedness values in grouping the extracted sentences in order to improve “story continuity” in the summary. For sentence realization, extractive summarization is used, where the summary sentences are picked from the source text.

Precision and recall of the extractive summary are evaluated by comparing it to original human written abstract. In general, a high *recall* means we have not missed much but we may have a lot of useless results. High *precision* means that the returned result was relevant, but that we might not have found all the relevant items.

$$precision = truepositives/(truepositives + falsepositives)$$

$$recall = truepositives/(truepositives + falsenegatives)$$

An example of true positives are the human written summary terms found in the extractive summary. False positives are the terms included in the extractive summary, but not found in the human written summary. False negatives mean the terms that were missed; those not found in the extractive summary, but which exist in the human written summary.

6.2 Results and discussion

We have constructed and evaluated summaries of four published scientific papers from the domain of speaker recognition (voice biometrics). The goal of the experimental part of this thesis is to act as a proof of concept. Our hypothesis was that high domain specificity, co-occurrence with highly domain specific content and frequent use of same (and semantically related) words indicate sentence extract-worthiness. Thus content of such sentences should be included in the summary.

Document	***Precision (best)***	***Recall (best)***	***Precision (worst)***	***Recall (worst)***
testDoc1.txt	0.438	0.814	0.290	0.581
testDoc2.txt	0.340	0.621	0.247	0.379
testDoc3.txt	0.427	0.889	0.369	0.472
testDoc4.txt	0.196	0.452	0.250	0.238

Table 14: Best and worst recall results: precision and recall of extractive summary generation for all test documents.

In general, a high *recall* means we have not missed much but we may have a lot of useless results. High *precision* means that the returned result was relevant, but that we might not have found all the relevant items. Our gold standard, a human written summary, may or may not contain all relevant information of the whole document. Because of this, we chose to accept low precision and high recall. This means that we focused on finding as much as possible of the relevant content and relying on the fixed limit of ten sentences in the extractive summary to prevent having too much irrelevant information to sift through. Table 15 shows results from different combinations of attributes.

Feature	***Precision***	***Recall***
DV	0.295	0.535
CO	0.419	0.605
SE	0.291	0.581
DV,CO	0.316	0.581
DV,SE	**0.471**	0.767
CO,SE	0.438	**0.814**
DV,CO,SE	0.397	0.581

Table 15: Results from experimenting with different combinations of attributes. The results show precision and recall of extractive summary generation for testdoc1 (Attachment 1). Features and feature groups in short format are domainVocabulary (DV), coOccurrence (CO) and semanticDistance (SE).

For comparison, we present results from Wong et.al (2008) in Table 16. The table shows classification performance based on different feature groups under the PSVM classifier. They worked with ROUGE datasets which contain 2000 labeled sentences and also used advanced methods like semi-supervised machine learning. Their work focused on getting high precision results. Wong et. al (2008) concluded that their most useful feature groups were surface and relevance, in other words, the external characteristics of a sentence in the document and the relationships of a sentence with other sentences in a cluster. Our results support their finding, that a combination of sentence characteristics and relatedness to other sentences produces best results. In our work the combinations of domain vocabulary and semantic relatedness and also co-occurrence and and semantic relatedness produced the best results.

Co-occurrence relevance measure improves summary quality

An obvious conclusion is that detecting domain vocabulary terms is essential for finding the domain-specific sentences. In addition to this, considering domain vocabulary co-occurrence data broadens the set of potentially extract-worthy sentences and thus acts as smoothing of the initial highly domain-specific set of sentences. Manual testing

Feature	***Precision***	***Recall***
Sur	0.488	0.146
Con	0.407	0.167
Rel	0.488	0.146
Event	0.344	0.146
Sur+Con	0.575	0.160
Sur+Rel	0.488	0.146
Con+Rel	0.588	0.139
Sur+Event	0.600	0.125
Con+Event	0.384	0.194
Rel+Event	0.543	0.132
Sur+Con+Event	0.595	0.153
Sur+Rel+Event	0.553	0.146
Con+Rel+Event	0.581	0.125
Sur+Con+Rel	0.595	0.174
Sur+Con+Rel+Event	0.579	0.153

Table 16: Results from other research. The table shows precisions and recalls of different feature groups under the PSVM classifier. The features used were surface (Sur), content (Con), relation (Rel) and event (Event) features.

with Weka machine learning framework from University of Waikato (Hall et. al, 2009) showed that when the categorization input parameters are considered alone, domain-Vocabulary parameter splits the input set of sentences roughly in to half (relevant / irrelevant), whereas coOccurrence and semanticDistance did better. They both tended to indicate that only one quarter of all sentences in the document are relevant.

Better results would naturally be achievable, both for domain specificity and relatedness measures, with a large domain-specific text corpora. Sets of documents from the same domain could be collected, for example, from online publisher databases. A similar approach has already been used, in WordNet::Similarity library, in collecting terms and calculating information content values for them from copyright protected dictionaries and thesauri.

Problems with hierarchical modeling

One of the hardest problems in computationally modeling meaning and structure of texts is the lack of world knowledge and the lack of simple means to represent world knowledge. There are, for example, no template situational models of events occurring in the world. FrameNet is a promising approach in resolving this problem, but at 2011 it still contains only a small set of template situations and is meant as training data for further machine learning. Other ontologies are sets of related terms, with little connection to the occurrence context of the contained terms.

Another problem is the lack of broader immediate context knowledge, an understanding of the domain of discourse. The enriching approach suggested in this paper does add context terminology, but not context template situations. And enriching with an external resource depends on the comprehensiveness of that resource; WordNet or any other resource only contain a limited set of common words.

7 Conclusions and future work

In this thesis we have discussed the sub-problems in discourse segmentation of texts: various NLP tasks of the source text, computing relevance, generality and relatedness of sentences and how to categorize the sentences to get a hierarchy and how to construct a summary of the whole text. We have presented a hierarchical and contextual approach to sentence topic modeling, where topic signatures were first generated for each sentence and the sentences were then categorized in order to find essential content. External resources, WordNet and information content data of words, were used in enriching and categorizing the topic signatures.

The proof of concept type experimental part of this thesis succeeded in constructing extractive summaries from published scientific papers. Our experiences with filtering the sentences in the experimental part show that collecting a domain vocabulary helps in finding the most domain-specific sentences from a text. Detecting co-occurrence with domain vocabulary terms broadens this set of extract-worthy sentences from the highly domain-specific ones. Sentence concept generality, calculated from sentence term information content values, is both useful in targeting the generated summary for the general public or the expert reader as well as in determining sentence relevance to the domain.

Comparing extractive summary to human written abstract resulted in approximately 50 percent term overlap. The bias of found terms was on the more domain specific ones. An extractive summary could be made more readable if sentence simplification was used, for example, by leaving out attribution clauses and initial adverbials. As such, the results are potentially useful to people working with document indexing, information extraction or message understanding.

The problem of extracting meaning and context from text is a broad topic, it is considered as one of the key problems on the path towards artificial intelligence (AI). This leaves room for many improvements. For example, topic signature relation detection could be improved with WSD, named entity recognition (NER) and statistical weighting of features.

Improvements to the algorithm

Many NLP methods were studied while writing this thesis, but only a few of them made it to the experimental part, mostly because of the broadness of the topic. Some improvements to the proposed algorithm include enriching sentence content with hypernyms, collecting related terms for word pairs and calculating how general terms the writer has used. The improvements should be executed after pre-processing the sentences, when dictionary forms of sentence words are available, but no calculations have yet been made.

- *Enrich with more general words:* Fetch direct hypernyms for each sentence word from WordNet and add them to sentence word vector. The goal is to improve sentence relatedness matching.
- *Collect context words:* Use the information content data precalculated for WordNet words to find most common words related to the sentence content. The goal is to improve sentence relatedness detection and content generality detection. Improvement steps include 1) collect information content values for sentence term pair least common subsumers from statistical occurrence data (WordNet information content files), 2) collect immediate hypernyms of sentence terms based on highest IC value, 3) add collected words to topic signature as immediate context words.
- *Calculate content generality:* Use the information content data precalculated for WordNet words to calculate the generality of sentence content. The goal is to enable targeting the extracted sentences to expert or novice readers. Improvement steps include 1) collect information content values of sentence terms (WordNet information content files) 2) pick k highest information content values from different sources and calculating average from those 3) categorize sentences to general (0-0,5) and specific (0,5-1) classes

References

Agirre, E., O. Ansa, D. Martinez, E.H. Hovy. 2001. Enriching WordNet Concepts with Topic Signatures. Proceedings of the Workshop on WordNet of the NAACL Conference. Pittsburgh, PA.

Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization (ISTS'97), 10-17, Madrid, 1997.

Doug Beeferman, Adam Berger, and John Lafferty. Text segmentation using exponential models. In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, 35-46, 1997.

J. Boyd-Graber, D. Blei. 2007. "PUTOP: turning predominant senses into a topic model for word sense disambiguation". In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), 277-281.

Bunnin, Nicholas and Jiyuan Yu (eds). The Blackwell Dictionary of Western Philosophy. Blackwell Publishing, 2004.

J. Cai, W. S. Lee, Y. W. Teh. 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.

Deerwester, S., et al, Improving Information Retrieval with Latent Semantic Indexing, Proceedings of the 51st Annual Meeting of the American Society for Information Science 25, 1988, 36-40.

Christiane Fellbaum, editor. 1998. WordNet: An Electronic Lexical Database. The MIT Press, Cambridge, MA.

Fillmore, Charles J. and Baker, Collin F. 2010. A Frame Approach to Semantic Analysis, in Heine, B. and Narrog, H. (eds.) Oxford Handbook of Linguistic Analysis, from OUP.

Gholamrezazadeh, Saeedeh; Salehi, Mohsen Amini; Gholamzadeh, Bahareh; , "A Comprehensive Survey on Text Summarization Systems," Computer Science and its

Applications, 2009. CSA '09. 2nd International Conference on , vol., no., 1-6, 10-12 Dec. 2009

Ralph Grishman, Beth Sundheim: Message Understanding Conference - 6: A Brief History. In: Proceedings of the 16th International Conference on Computational Linguistics (COLING), I, Kopenhagen, 1996, 466-471.

Barbara J. Grosz and Candance L. Sidner. 1986. Attention, intentions, and the structure of discourse. Computational Linguistics, 12:175-204.

Hart, P. E.; Nilsson, N. J.; Raphael, B. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". IEEE Transactions on Systems Science and Cybernetics SSC4 4 (2): 100-107

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

M. A. K. Halliday and Ruqaiya Hasan. Cohesion in English. Longman, London, 1976.

Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proc 14th Conference on Computational linguistics. (1992) 539-545

Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. In ACL 32, 9-16.

Graeme Hirst and Alexander Budanitsky. 2001. Lexical Chains and Semantic Distance. University of Toronto. Eurolan-2001, August 2001, Iasi, Romania

Michael Hoey. 1991. Patterns of Lexis in Text. Oxford University Press.

William A. Hollingsworth. Using Lexical Chains to Characterise Scientific Text. PhD thesis, Clare Hall College, University of Cambridge, 2008.

Jarmasz, M., Szpakowicz, S.: The design and implementation of an electronic lexical knowledge base. In: Proc 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence (AI 2001). (2001) 325-334

Jarmasz, M. and Szpakowicz, S. (2003a). Roget's Thesaurus and Semantic Similarity. N. Nicolov, K. Bontcheva, G. Angelova, R. Mitkov (eds.) Recent Advances in Natural

Language Processing III: Selected Papers from RANLP 2003, John Benjamins, Amsterdam/Philadelphia, Current Issues in Linguistic Theory (CILT), vol. 260, 111-120.

Jarmasz, M. and Szpakowicz, S. (2003b). Not As Easy As It Seems: Automating the Construction of Lexical Chains Using Roget's Thesaurus. Proceedings of the 16th Canadian Conference on Artificial Intelligence (AI 2003), Halifax, Canada, June, 544-549.

Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of International Conference on Research on Computational Linguistics (ROCLING X), Taiwan, 1997.

Jurafsky, D., Martin,J.H. (2009). Speech and Language Processing. New Jersey: Prentice Hall

Kawtrakul A, Yingsaeree C. (2005), A Unified Framework for Automatic Metadata Extraction from Electronic Document, Proceedings of IADLC2005 (The International Advanced Digital Library Conference) (25-26 August 2005), 71-77.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, 423-430.

Hideki Kozima and Teiji Furugori. 1993. Similarity between words computed by spreading activation on an English dictionary. In Proceedings of 6th Conference of the European Chapter of the Association for Computational Linguistics (EACL-93), 232-239, Utrecht, 1993.

Hideki Kozima and Akira Ito. 1997. Context-sensitive word distance by adaptive scaling of a semantic space. In Ruslan Mitkov and Nicolas Nicolov, editors, Recent Advances in Natural Language Processing: Selected Papers from RANLP 95, volume 136 of Amsterdam Sudies in the Theory and History of Linguistic Science: Current Issues in Linguistic Theory, chapter 2, pages 111-124. John Benjamins Publishing Company, Amsterdam/Phildadelphia, 1997.

Lin, C.-Y. and E.H. Hovy. 2000. The Automated Acquisition of Topic Signatures for Text Summarization. Proceedings of the COLING Conference. Saarbrücken, Germany.

Christopher Manning. 1998. Rethinking text segmentation models: An information extraction case study. Technical report SULTRY-98-07-01, University of Sydney.

Manning, Christopher D. and Hinrich Schütze. 1999. "Foundations of Statistical Natural Language Processing". Cambridge, MA: MIT Press.

George A. Miller. 1990. WordNet: An on-line lexical database. International Journal of Lexicography, 3(4):235-312. Special Issue.

George A. Miller and Walter G. Charles. 1991. Contextual correlates of semantic similarity. Language and Cognitive Processes, 6(1):1-28.

Miltsakaki, Eleni and Karen Kukich. 2000. The role of centering theory's rough-shift in the teaching and evaluation of writing skills. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, 408-415, Hong Kong.

Mohammad, S., Hirst, G.: Distributional measures of concept-distance: A task-oriented evaluation. In: Proceedings, 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), Sydney, Australia (July 2006)

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics, 17(1):21-48, March 1991.

Nakamura, J., Nagao, M.: Extraction of semantic information from an ordinary english dictionary and its evaluation. In: Proc 12th Conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1988) 459-464

Pantel, P., Pennacchiotti, M.: Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In: Proc 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, Association for Computational Linguistics (July 2006) 113-120

Pantel, P., Ravichandran, D.: Automatically labeling semantic classes. In: Proc 2004 Human Language Technology Conference (HLT-NAACL-04). (2004) 321-328

Rebecca J. Passonneau and Diane J. Litman. 1993. Intention-based segmentation: Human reliability and correlation with linguistic cues. In ACL 31, 148-155.

Pedersen, T., Patwardhan, S., and Michelizzi J.: WordNet::Similarity - Measuring the Relatedness of Concepts, Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04), 38-41, May 3-5, 2004, Boston, MA. (Demonstration System)

Putnam, H. 1975. The meaning of "meaning". In K. Gunderson (Ed.), Language, Mind and Knowledge, Vol. VII of Minnesota Studies in the Philosophy of Science, 131-193. University of Minnesota Press.

Peter M. Roget.: Roget's Thesaurus of English Words and Phrases, 1911. (http://www.gutenberg.org/ebooks/10681)

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. Communications of the ACM, 8(10):627-633, October 1965.

Gerard Salton, J. Allan, and Chris Buckley. Approaches to passage retrieval in full text information systems. In SIGIR 1993: Proceedings of the 16th Annual International 47 ACM/SIGIR Conference on Research and Development in Information Retrieval, 49-58, Pittsburgh, PA, USA, 1993. ACM.

Snow, R., Jurafsky, D., Ng, A.Y.: Learning syntactic patterns for automatic hypernym discovery. In Saul, L.K., Weiss, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, MA (2005) 1297-1304

Heather Stark. What do paragraph markers do? Discourse Processes, 11(3):275-304, 1988.

Weeds, J.E.: Measures and applications of lexical distributional similarity. PhD thesis, University of Sussex (September 2003)

Wong, K. F., Wu, M., and Li, W. 2008. Extractive summarization using supervised and semi-supervised learning. In Proceedings of the International Conference on Computational Linguistics (COLING'08), 985-992.

X. Zhang, X. Zhou, and X. Hu, "Semantic Smoothing for Model-based Document Clustering," in 2006 IEEE International Conference on Data Mining (IEEE ICDM06), Dec. 18-22, 2006, HongKong, 1193-1198

X. Zhou, X. Hu, and X. Zhang, "Topic Signature Language Model for Ad hoc Retrieval," IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 19, no. 9, 1276-1287, Sept., 2007

Attachment 1: Algorithm input data

Text-independent speaker recognition using graph matching.

Technical mismatches between the training and matching conditions adversely affect the performance of a speaker recognition system. In this paper, we present a matching scheme which is invariant to feature rotation, translation and uniform scaling. The proposed approach uses a neighborhood graph to represent the global shape of the feature distribution. The reference and test graphs are aligned by graph matching and the match score is computed using conventional template matching. Experiments on the NIST-1999 SRE corpus indicate that the method is comparable to conventional Gaussian mixture model (GMM) and vector quantization (VQ)-based approaches.

One of the biggest challenges in automatic speaker recognition is obtaining invariance across varying operating conditions, while retaining maximum speaker variability. Different handset type, transmission line/coding, and background noise are typical factors, which lead to signal mismatch across training and recognition. For a speaker recognition system to be useful in practice it needs to be optimized against the mismatch problem. Various approaches have been proposed for tackling the invariance problem, including robust feature extraction (Mammone et al., 1996), feature normalization (Pelecanos and Sridharan, 2001), model transformation (Kenny et al., 2007; Teunen et al., 2002; Vogt and Sridharan, 2008), and match score normalization (Auckenthaler et al., 2000; Reynolds et al., 2000). State-of-the-art text-independent speaker recognizers use mean subtraction at the utterance level, often referred to as cepstral mean subtraction (CMS) in the context of cepstral features. The assumption in mean subtraction is that all the feature vectors have been translated by an unknown channel-dependent vector. By subtracting the mean from both the training and testing vectors, the matching is less affected by this bias. For clean data (no channel mismatch), CMS degrades accuracy. A general affine channel/environment model (Mak and Tsang, 2004; Mammone et al., 1996) includes rotation and scaling of the feature vectors in addition to the additive bias. The three transformations - rotation, scaling, and translation - can be collectively expressed as an affine feature distortion model: y $\frac{1}{2}$ Ax þ b. The matrix A and vector b are channel-dependent transformation parameters, whereas x and y are the clean and the noisy (observed) vectors, respectively. In image- and video-based biometrics, invariance against rotation, translation and scaling is often desirable. For instance, a face recognizer would produce the same match score, independent of face tilting (rotation), location with respect to the background (translation) and distance from the camera (scale). A natural idea to achieve invariance is to construct a graph from certain feature points from the images and then to use graph matching (Bunke and Shearer, 1998) methodology. In the matching phase, only the graph structures - and not the original feature points - are compared. For example, Burge and Burger (2000) use Voronoi diagram graphs to model ear shape. The graphs of the reference ear and the unknown ear were matched using error-correcting graph matching. It is an open question whether similar ideas could be adopted to speaker recognition. In our view, formulation of a transformation invariant matching scheme for speech features poses several challenges. First, images are two-dimensional, and the semantic meaning of the constructed graph can be visually verified. However, commonly used speech spectrum features are high-dimensional (10-40 dimensions), and it is difficult to give an intuitive meaning to the graph calculated from the extracted features. Second, in text-independent speaker recognition, the feature distributions of the training vectors and the test utterance are likely to vary because of text mismatch in addition to the technical mismatches mentioned above. It is also unclear whether the matching should use the whole distributions, or should a good match be indicated if the sub-graphs from the reference vectors and the test utterance match well. The motivation of this paper is to experiment with a few simple ideas. To our knowledge, no graph-based matching has previously been proposed for text-independent speaker recognition. The overview of the proposed scheme is illustrated in Fig. 1. We first cluster both the training and the testing vectors into a small number of clusters, represented by a set of centroid vectors. Neighborhood graphs are then constructed for both sets. Finally, structural similarity of the reference and the test graphs is evaluated by calculating the degree of isomorphism between the graphs. We also propose a matching framework which is a hybrid between graph-based structural matching and vector-based template matching. Graph matching is used as a pairing tool between the reference and the test centroids. The paired vectors from each set are then used for finding the parameters of the affine transformation model. Finally, the match score is computed as the distortion between the compensated centroids. Feature and speaker model transformations, including the affine transformation, have been studied by different authors (Kenny et al., 2007; Mak and Tsang, 2004; Mammone et al., 1996; Siohan and Lee, 1997; Vogt and Sridharan, 2008). These methods usually require either parallel training data recorded simultaneously through various handsets, or a large number of training utterances collected from multiple recording sessions from a number of speakers. These datasets are then used for estimating the transformation parameters. The method that we propose, in turn, aligns the test vectors to the claimed speaker's model during verification. Therefore, the proposed method does not require any external data or training of a channel/session variability model. The rest of the paper is organized as follows. In Section 2, we give details of the structural graph matching framework. Section 3 describes the hybrid structural and template matching algorithm. Experimental setup and the results are described in Section 4. Finally, conclusions are drawn in Section 5.

In this study, we have introduced graph-based matching approach to text-independent speaker recognition. The approach was motivated by the fact that a neighborhood graph encodes structural information about the feature space. Under the affine distortion model - including rotation, translation, and uniform scaling - ideally the neighborhood graph should not change. The performance of the proposed method was comparable to the GMM- and VQ-based approaches. A fusion experiment demonstrated that GMM- and graph-based methods might contain mutually complementary information. The approach has potential to complement or replace currently used statistical and templatebased methods. The method, however, has several practical problems to be solved before it can be utilized in real-life speaker recognition systems. First, exact graph matching is computationally expensive, and heuristic algorithm needs to be used which weakens the performance. Second, the size of the association graph grows fast for large models, which implies increased running time. The largest graph that we could test in reasonable time was 128. A possible future solution could be based on a heuristic algorithm, which solves the graph matching problem directly, without reducing it first to the maximum clique search from the association graph. To further speed up scoring in the identification task, some form of decision tree in which the feature points represent tree nodes, could be used. In the current approach, the feature points from the reference and test sets were found by clustering and implicitly assumed to correspond to phonetic classes. In general, the joint effects of channel and text (phonetic) mismatch are not well understood. Recently, excellent results have been obtained by using phone-class constrained GMMs which reduces text mismatch by phone recognition (Castaldo et al., 2007). The graph-based method could be used by restricting matching onto the same phone classes between training and test utterances. Different graph structures are also possible. In this study, we considered unweighted kNN graph where a node is either connected or not to another node. A possible future direction would be using real-valued weights (such as Euclidean distances between points) and re-defining the matching framework for such graphs. Current likelihood-based (or frame-based) approaches also assume independence of the frames, largely ignoring utterance-level structural information. Graph matching could be potentially used as an alternative matching tool for the existing GMM-based systems. These are points for future research.

Attachment 2: Categorization results

For testdoc1, categorization marked 25 out of 70 sentences as relevant. Class attribute value 1 indicates relevant, 0 irrelevant.

```
@relation sentenceRelevance

@attribute domainVocabulary numeric
@attribute coOccurrence numeric
@attribute semanticDistance numeric
@attribute class numeric

@data
0,0.2,0.017991,0
0,0.25,0.114542,0
0,0.3,0.224857,1
0.333333,0.4,0.29816,1
0.333333,0.55,0.424506,1
1,0.65,0.460848,1
0.333333,0.5,0.353678,1
0,0.2,0.531935,0
0,0.2,0.267282,0
0.333333,0.7,0.990051,1
0.666667,0.45,0.197794,1
0,0.35,0.268622,1
0,0.15,0.251559,0
0.333333,0.25,0,1
0,0.2,0.223981,0
0.333333,1,1,1
```

```
0,0.25,0.080932,0
0.666667,0.7,0.557194,1
0,0.25,0.674107,0
0,0.15,0.33177,0
0.333333,0.35,0.231713,1
0,0.25,0.275169,0
0,0.15,0.176916,0
0,0.2,0.246404,0
0,0.1,0.209805,0
0,0.2,0.608794,0
0,0.4,0.722254,0
0,0.45,0.572813,0
0,0.05,0.148771,0
0.333333,0.4,0.135059,1
0.333333,0.25,0.06923,1
0,0.05,0.255013,0
0,0.05,0.172999,0
0,0.15,0.285479,0
0,0.2,0,0
0,0.35,0.220939,1
0,0.25,0.302077,0
0.333333,0.3,0.107892,1
0,0.3,0.333832,0
0,0.15,0.475334,0
0,0.15,0.096861,0
0,0.2,0.386309,0
0.333333,0.4,0.185886,1
0,0,0.05557,0
0,0.05,0.209805,0
0,0,0.000567,0
0,0,0.131038,0
0,0,0.172844,0
0,0.35,0.130213,1
0,0.25,0.457704,0
0,0.35,0.246868,1
```

```
1,0.35,0.070622,1
0.333333,0.4,0.066034,1
0.333333,0.25,0.095984,1
0,0.25,0.291561,0
0,0.25,0.198155,0
0,0.05,0.474612,0
0.333333,0.15,0.089592,1
0,0.2,0.570751,0
0,0.15,0.809887,0
0,0.15,0.600804,0
0,0.1,0.124543,0
0,0.2,0.342698,0
0,0.35,0.360792,0
0,0.05,0.074179,0
0.666667,0.3,0.144028,1
0,0.15,0.427909,0
0,0.2,0.159183,0
0.333333,0.4,0.149853,1
0,0,0.241868,0
```

Attachment 3: Constructed extractive summaries

testDoc1.txt extractive summary

In this paper , we present a matching scheme which is invariant to feature rotation , translation and uniform scaling . The reference and test graphs are aligned by graph matching and the match score is computed using conventional template matching . One of the biggest challenges in automatic speaker recognition is obtaining invariance across varying operating conditions , while retaining maximum speaker variability . Various approaches have been proposed for tackling the invariance problem , including robust feature extraction , feature normalization , model transformation , and match score normalization . For clean data no channel mismatch , CMS degrades accuracy . The three transformations rotation , scaling , and translation can be collectively expressed as an affine feature distortion model : y Ax b. The matrix A and vector b are channel dependent transformation parameters , whereas x and y are the clean ' and the noisy ' observed vectors , respectively . Graph matching is used as a pairing tool between the reference and the test centroids . Finally , the match score is computed as the distortion between the compensated centroids . In this study , we have introduced graph based matching approach to text independent speaker recognition . Under the affine distortion model including rotation , translation , and uniform scaling ideally the neighborhood graph should not change .

testDoc2.txt extractive summary

Temporally Weighted Linear Prediction Features for Tackling Additive Noise in Speaker Verification . In the popular mel frequency cepstral coefficient MFCC front end , the conventional Fourierbased spectrum estimation is substituted with weighted linear predictive methods , which have earlier shown success in noiserobust speech recognition . Two temporally weighted variants of linear predictive modeling are introduced to speaker verification and they are compared to FFT , which is normally used in computing MFCCs , and to conventional linear prediction . The new features hold a promise for noiserobust speaker verification . The standard spectrum analysis method for speaker verification is the discrete Fourier transform , implemented as the fast Fourier transform FFT . Research in speaker recognition over the past two decades has largely concentrated on tackling the channel variability problem , that is , how to normalize the adverse effects due to differing training and test handsets or channels e.g. GSM versus landline speech . Another approach to increase robustness is to carry out feature normalization such as cepstral mean and variance normalization CMVN , RASTA filtering or feature warping . All these investigations , however , use vector quantization VQ classifiers and some of the feature extraction methods utilized are computationally intensive , because they involve solving for the roots of LP polynomials . Differently from these previous studies , this work a compares two straightforward noise robust modifications of LP and b utilizes them in a more modern speaker verification system based on adapted Gaussian mixtures and MFCC feature extraction . The robust linear predictive methods used for spectrum estimation Fig . 1 are weighted linear prediction WLP and stabilized WLP SWLP , which is a variant ofWLP that guarantees the stability of the resulting allpole filter .

testDoc3.txt extractive summary

Multitaper Estimation of Frequency Warped Cepstra With Application to Speaker Verification . Usually the mel frequency cepstral coefficients are estimated either from a periodogram or from a windowed periodogram . HE cepstrum was introduced by Bogert , Healy and Tukey in the early 1960s . In these applications , a psycho acoustically motivated frequency warping transformation is usually applied to the spectrum before the logarithm and the inverse Fourier transform , such as in the popular mel frequency cepstral coefficients MFCCs . The periodogram suffers from large bias and large variance , altogether causing large estimation errors in the cepstral coefficients . The windowed periodogram has low bias in general , but it still suffers from high variance . The multitaper spectrum estimator is known to have low variance , but has not gained much attention in MFCC estimation . Finally , we demonstrate the effectiveness of multitaper MFCC estimation over the conventional Hamming window based MFCC extraction , in a speaker verification context . The result was the same for the phoneme l , indicating the robustness of the multitaper estimator for speech like processes . We also demonstrated that the peak matched MFCC performs slightly better than the Hamming windowMFCCin the NIST 2006 SRE .

testDoc4.txt extractive summary

In addition to these physical differences , each speaker has his or her characteristic manner of speaking , including the use of a particular accent , rhythm , intonation style , pronounciation pattern , choice of vocabulary and so on . An important application of speaker recognition technology is forensics . Speaker diarization , also known as

’ who spoke when ’ , attempts to extract speaking turns of the different participants from a spoken document , and is an extension of the ’ classical ’ speaker recognition techniques applied to recordings with multiple speakers . In forensics and speaker diarization , the speakers can be considered non cooperative as they do not specifically wish to be recognized . Textindependent recognition is the much more challenging of the two tasks . In general , any variation between two recordings of the same speaker is known as session variability . In addition , we give emphasis to the recent techniques that have presented a paradigm shift from the traditional vector based speaker models to so called supervector models . Section 6 is then devoted to the current supervector classifiers and their session compensation . In Section 7 we discuss the evaluation of speaker recognition performance and give pointers to software packages as well . We have presented an overview of the classical and new methods of automatic text independent speaker recognition .

Attachment 4: Precision and recall of extractive summarization

testdoc1

humanSummaryWords : (43) [GMM, NIST-1999, SRE, Technical, VQ, affect, align, approach, base, be, condition, corpus, distribution, experiment, feature, graph, indicate, match, matching, method, mismatch, mixture, model, neighborhood, paper, performance, present, quantization, recognition, reference, represent, rotation, scaling, scheme, score, shape, speaker, system, test, training, translation, use, vector]
extractedSummaryWords: (63) [Ax, CMS, accuracy, align, approach, b, b., base, be, centroid, challenge, change, channel, compensate, compute, condition, datum, degrade, distortion, express, extraction, feature, graph, have, include, introduce, invariance, match, matching, matrix, mismatch, model, neighborhood, noisy, normalization, obtain, operating, pairing, paper, parameter, present, problem, propose, recognition, reference, retain, rotation, scaling, scheme, score, speaker, study, tackle, test, text, tool, transformation, translation, use, variability, vary, vector, y]

misses : 18 out of 43 human summary words not found in extractive summary
falsePositives : 38 out of 63 extractive summary words not found in human summary

————————————-

precision : 25/63 = 0.3968254 = 39.68254%
recall : 25/43 = 0.5813953 = 58.139534%

testdoc2

humanSummaryWords : (58) [%, FFT, MFCC, MFCCs, NIST, SNR, SRE, accuracy, author, baseline, be, coefficient, compare, compute, consider, corpus, corruption, datum, db, eer, effect, enhancement, estimation, experiment, factory, feature, front-end, give, have, hold, improve, include, indicate, introduce, investigate, level, mel-frequency, method, modeling, noise, performance, prediction, preprocessing, promise, propose, recognition, representation, show, speaker, spectrum, speech, substitute, subtraction, success, system, use, variant, verification]
extractedSummaryWords: (99) [Additive, CMVN, FFT, Features, Fig, Fourier, GSM, LP, Linear, MFCC, MFCCs, Noise, Prediction, RASTA, Research, SWLP, Speaker,

Tackling, Temporally, VQ, Verification, WLP, Weighted, adapt, analysis, approach, b, base, be, carry, channel, classifier, coefficient, compare, compute, concentrate, decade, differ, e.g., effect, end, estimation, extraction, feature, filter, frequency, guarantee, handset, have, hold, implement, increase, introduce, investigation, involve, landline, mean, mel, method, mixture, modeling, modification, noise, normalization, normalize, ofWLP, polynomial, prediction, problem, promise, quantization, recognition, result, robustness, root, show, solve, speaker, spectrum, speech, stability, stabilize, study, substitute, success, system, tackle, test, training, transform, use, utilize, variability, variance, variant, vector, verification, warping, work]

misses : 29 out of 58 human summary words not found in extractive summary
falsePositives : 70 out of 99 extractive summary words not found in human summary

———————-

precision : 29/99 = 0.2929293 = 29.292929%
recall : 29/58 = 0.5 = 50.0%

testdoc3

humanSummaryWords : (36) [??, Carlo, Hamming, Monte, NIST, approximation, be, bias, coefficient, compare, computation, context, demonstrate, error, estimate, estimator, formula, have, include, match, mean, mel-frequency, peak, perform, periodogram, process, propose, show, speaker, square, state, task, use, variance, verification, window]
extractedSummaryWords: (65) [1960, Application, Bogert, Cepstra, Estimation, Fourier, Frequency, Hamming, Healy, MFCC, Multitaper, NIST, SRE, Speaker, Tukey, Verification, Warped, , application, apply, attention, base, be, bias, cause, cepstrum, coefficient, context, demonstrate, effectiveness, error, estimate, estimation, estimator, extraction, frequency, gain, hamming, have, indicate, introduce, know, l, logarithm, match, mel, mfcc, peak, perform, periodogram, phoneme, process, result, robustness, speaker, spectrum, speech, suffer, transform, transformation, variance, verification, warp, window, windowMFCCin]

misses : 16 out of 36 human summary words not found in extractive summary
falsePositives : 45 out of 65 extractive summary words not found in human summary

———————-

precision : 20/65 = 0.30769232 = 30.769232%
recall : 20/36 = 0.5555556 = 55.555557%

testdoc4

humanSummaryWords : (42) [Speaker, address, area, be, concern, conclude, decade, development, direction, discuss, discussion, elaborate, emphasis, evaluation, exploration, extraction, feature, fundamental, give, have, method, methodology, modeling, open, overview, paper, progress, provide, recognition, represent, robustness, session, speaker, start, study, supervector, system, technique, technology, trend, variability, vector]
extractedSummaryWords: (66) [Speaker, Textindependent, accent, addition, application, apply, attempt, base, be, call, choice, classifier, compensation, consider, cooperative, devote, diarization, difference, discuss, do, document, emphasis, evaluation, extension, extract, forensic, give, have, include, intonation, know, manner, method, model, overview, package, paradigm, participant, pattern, performance, pointer, present, pronounciation, recognition, recognize, recording, rhythm, section, session, shift, software, speak, speaker, speaking, style, supervector, task, technique, technology, text, turn, use, variability, variation, vector, wish]

misses : 25 out of 42 human summary words not found in extractive summary
falsePositives : 49 out of 66 extractive summary words not found in human summary
——————-
precision : 17/66 = 0.25757575 = 25.757576%
recall : 17/42 = 0.4047619 = 40.476192%