



Norwegian University of
Science and Technology

Extracting Keyphrases from Individual News Articles

Kristian Lund

Master of Science in Computer Science

Submission date: June 2011

Supervisor: Jon Atle Gulla, IDI

Co-supervisor: Stein L. Tomassen, IDI
Aleksander Øhrn, CXense

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

Extraction of keyphrases from individual documents is a research area in which one try to extract a small set of keyphrases that describe the content of a single document. The advantages with this form of extraction is that it retains most of the semantic context from the document.

In this thesis we focus on the news article domain and use the structure of a news article to improve the quality of the extracted keyphrases. An existing individual document keyphrase extraction algorithm is used as the basis. This algorithm is enhanced by implementing a weighting system based upon the structure of news articles. In addition some other common methods for keyword extraction is applied. The effects of these changes are tested extensively in the evaluation phase.

In the evaluation of the implemented prototype we find that the introduction of a weight based system yields results that are equal to the basic algorithm and that few improvements can be made. We do however find that an automatically generated stopword list based on the corpus improves the results by 1-2%.

Preface

This report is a documentation of the project work performed in the course TDT4520 “Computer and Information Science, Master Thesis” by Kristian Lund. The project counts for 30 units and is carried out in the tenth semester of the Master of Science education in Computer Science at The Norwegian University of Technology and Science, NTNU.

I wish to thank my supervisor Jon Atle Gulla at the Department of Computer and Information Science, NTNU for valuable feedback and advice. I also wish to thank my co-supervisor Aleksander Øhrn from cXense for the provided data and the advice. Finally I wish to thank my co-supervisor Stein L. Tomassen for his continued advice, providing of data and support throughout the semester.

Trondheim, June 6, 2011

Kristian Lund

Contents

I	Introduction	1
1	Introduction	2
1.1	Problem	2
1.2	Approach	3
1.3	Report Structure	3
2	Research Approach	5
2.1	Introduction	5
2.2	Methodology	6
2.3	Overall Research Phases	10
II	Theoretical Background	12
3	Theoretical Overview	13
3.1	Text Categorization	13
3.2	Keyword Extraction	15

3.3	News Articles Domain	15
3.4	Individual Document Extraction vs. Corpus Based Ex- traction	16
3.5	Term Selection Approaches	17
3.5.1	N-Grams	17
3.5.2	Noun Phrase Chunking	17
3.5.3	Part-of-Speech Tag Patterns	18
3.5.4	Stopword Limited Phrases	18
3.5.5	Graph Based Co-Occurrence	18
3.5.6	Syntactic Filter	18
4	Related Work	19
4.1	Machine Learning	19
4.1.1	Kea and Kea++	19
4.1.2	Hulth	20
4.2	Other Approaches	21
4.3	Individual Document Extraction	22
4.3.1	TextRank	22
4.3.2	RAKE	24
4.3.3	Multilingual Single Document Keyword Extrac- tion	25

III	Prototype Implementation	28
5	Approach	29
5.1	Constraints	29
5.2	Comparison of Individual Document Keyphrase Extraction Methods	30
5.3	Overall Architecture	32
5.4	Stopword List	33
6	Implementation	36
6.1	HTML Parsing	36
6.2	Keyphrase Extraction	40
6.2.1	Candidate Extraction	42
6.2.2	Wordscore Calculation	43
6.2.3	Noun Phrase Filter	44
6.2.4	Phrase Score Calculation	44
6.2.5	Weighting	45
6.2.6	Stopword Trigrams	46
6.3	Clustering	46
IV	Evaluation and Conclusion	48
7	Evaluation	49
7.1	The Evaluation Process	49

7.2	The Classification Algorithms	51
7.3	The Corpus	51
7.4	Methods	52
7.5	Results	53
8	Conclusion	56
9	Future Work	58
9.1	Multi-label Classification	58
9.2	Using Keyphrases to Enhance Existing Systems	59
9.3	Classification System with Semantic Relatedness	59
A	Stopword List	67
A.1	List Generated with Odds Ratio	67
A.2	Stopword Pre-Defined List	68
B	Groups Used in Evaluation	69
C	External Resources	70
C.1	Jericho HTML Parser	70
C.2	Oslo-Bergen-Tagger	71
C.3	Norsk Ordbank	71

Part I

Introduction

Chapter 1

Introduction

In natural language processing a common theme has for many areas been the amount of semantic context that can be used. Individual document keyphrase extraction tries to maintain the important semantic context from the document while also identifying those terms or phrases that are describing for the document. In this thesis the field of interest is the news articles domain. News articles have some special traits that separates them from standard documents and we look at how these special traits can be exploited to improve the performance of individual document keyword extraction.

1.1 Problem

Extraction of the most important information from a document is a field of great interest. The results of this extraction can be used in many different areas including document keyword assignment and document classification. Traditionally the information from documents have been extracted with statistical methods using the statistics from

the entire corpus to determine the most important words. The problem with this approach is that most of the semantic context is lost. By instead extracting phrases of words from each individual document it is possible to retain most of the semantic context.

The purpose of this project is to use individual document keyphrase extraction to find phrases from each individual document that is describing the content of the document. The phrases should also be discriminating towards other documents. The language for the documents used in the project will be Norwegian.

1.2 Approach

The approach to this project is divided in four steps:

1. An introduction where the problem is defined and a research approach is constructed.
2. The building of a theoretical foundation.
3. Implementation of a prototype system.
4. Evaluation of the system.

A detailed description of the different research phases is found in chapter 2.

1.3 Report Structure

Chapter 2: Research Approach The research method and methodology used in this thesis are described.

Chapter 3: Theoretical Overview In this chapter the methods and theories that are used in this thesis is explained. Some discussion of the domains used in the thesis is also done.

Chapter 4: Related Work The works from related projects are presented in this chapter. These projects are also compared and what methods that should be used in this thesis is discussed.

Chapter 5: Approach The approach used for realizing the desired qualities in the system is described.

Chapter 6: Implementation This chapter gives a detailed description of the implementation of the system and describes the architecture.

Chapter 7: Evaluation An evaluation of the constructed system to estimate the performance is conducted. The evaluation consists of using classifiers to compare the results from this system to the results of a state-of-the-art algorithm.

Chapter 8: Conclusion This chapter presents concluding remarks for the work done.

Chapter 9: Future Work We discuss the possibilities for further work with the system.

Chapter 2

Research Approach

This chapter presents the overall research approach.

2.1 Introduction

According to [36] research has traditionally been divided into two categories: pure- and applied-research. Pure research construct theories while applied research test the theories in the real world. They find this twofold classification too restrictive and that it does not very well reflect the research applied in academic disciplines. They proposed a classification of research in three types: exploratory, testing-out, and problem solving.

Exploratory Research This type of research seeks to solve a new problem where little information is available. This causes the research ideas to be unclear and the research will have to examine if existing methodologies can be used and if current theories and concepts can be applied.

Testing-out Research This research test the limitations of previous proposed generalizations. It seeks to answer under what conditions the theory applies.

Problem-solving Research A particular real world problem is used as the starting point for this kind of research. The research starts by formulating the problem and the goal is to discover a method of solution. This kind of research will usually involve many different theories and methods that can span across several academic disciplines.

The research conducted in this project is most closely related to testing-out research. It does however have some elements that are related to the the two other types of research. The documents are in Norwegian and little research has been conducted in the field of Norwegian text processing. This reveals the similarities with exploratory research. The problem was defined in section 1.1 as how single document keyword extraction can be used for news article classification. This is a real world setting and hence the research has elements of problem-solving research as well. The primary part of this project is however is to use an existing general algorithm and apply it to a new field. This field is the extraction of keyphrases from news articles and the application of a general algorithm to test the results for a more specific field is a form of testing-out research.

The approaches and methodologies that were used are presented in the following sections.

2.2 Methodology

In this project the design-science methodology is used. The design-science methodology is a very general methodology and was described

by Hevner et al.[37]. It is designed as a problem solving methodology and is therefore a useful tool for most research projects. The paper picks up the work that was started by Heinz and Myers[38]. They stated that: This paper can be seen as a response to the call “to discuss explicitly the criteria for judging qualitative, case and interpretive research in information systems” [39]. Therefore the methodology is constructed to give an output where it is possible to give an explicit evaluation and hence makes it possible to establish the quality of the research and communicate the results to audiences.

The methodology has seven guidelines that should help in the designed and they are summarized in Table 2.1.

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Table 2.1: Design-Science Research Guidelines [37]

- **Guideline 1** requires that an artifact is produced during the project. In this project the artifact will be the weighted RAKE system.

- **Guideline 2** demands that the project should produce a solution that can be applied to relevant business problems. Extraction of information from documents is used for both search engines and classification systems. Better information extraction can therefore produce better results for both systems. The area of primary focus in this project is to extract terms from the documents that can improve the results of classification.
- **Guideline 3** demands that the artifact that is produced should be evaluated with well known and accepted evaluation methods to determine the results of the research. The project is evaluated by using the extracted terms and use the results to perform classification of the documents. The results from this classification determines the value of results.
- **Guideline 4** stress the need for the application to be a contribution to the area it was designed for. To be a contribution an artifact should either solve an unsolved problem or it should solve a problem with an existing solution in a more effective way. In this project we attempt to solve a problem in a more effective way. The problem is extraction of terms from documents and we try to extract terms that better describe the content of the document than current methods.
- **Guideline 5** informs the reader that rigorous methods should be used. The methods used in this project can all be traced back to previously conducted work and therefore they have a solid theoretical foundation. The evaluation measures the effects of the proposed methods and gives concrete data on their performance.
- **Guideline 6** requires that the design should be considered as a search process. This will inherently become iterative as a search

for the optimal solution is often intractable in realistic systems. Effective design will require knowledge about both the application domain and the solution domain[16]. Therefore the process becomes iterative since better design decisions can be made with more knowledge. In this project we first search for knowledge and ideas in similar projects. Then an approach is defined before the solution is implemented. If any problems occur or adjustments should be made based upon new knowledge we can go back to the previous step in the process.

- **Guideline 7** demands that the results are presented effectively to all audiences that have an interest in the result. To present something effectively it should be easy to understand and therefore the results from this project is presented in this report as well as the background and prestudy. This should give the reader sufficient information to understand the material that is presented in the report.

2.3 Overall Research Phases

The research in this project was divided into four phases and the research progression is depicted in figure 2.1.

The first phase consisted of analyzing related projects. The methods used where examined and the strenghts and weaknesses of these methods where discovered. This phase is described by chapter 3 and 4.

In the second phase an approach is defined. Based upon the analysis from phase one some methods are chosen for implementation and the system architecture is designed. The results from this phase are described in chapter 5.

The third phase in this project is the implementation of the system. Detailed design and coding is performed in this part of the project. It is described in chapter 6.

The fourth and final phase of the project is the evaluation. The most natural way to evaluate the output from this system would be to compare the output to manually assigned keywords. Unfortunately to the authors knowledge there exists no document collection with manually assigned keywords for Norwegian. Therefore a classifier is used to evaluate the results from the system. The evaluation is described in chapter 7.

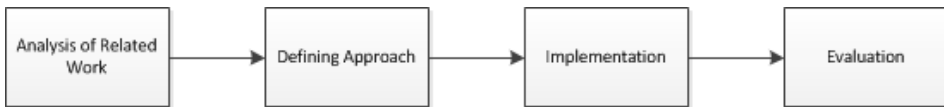


Figure 2.1: Research Phases

Part II

Theoretical Background

Chapter 3

Theoretical Overview

In this chapter the theoretical background for the project is described. Some key concepts are described and we look at some of the contexts where our work can be used. We also look at opportunities and constraints that will be present in the news article domain where we expect the system to be used.

3.1 Text Categorization

Text categorization is to identify what category a document belongs to. Traditionally the knowledge engineered approach was used[3], which consists of manually created rules for each category. In the '90s the machine based learning approach became more popular. This approach trains an algorithm on a training set and the algorithm can then classify documents automatically. This approach saves time for expert human labor that do not have to create rules for all categories and the approach can obtain results that are comparable to that of human experts.

Text categorization will usually consist of at least two steps. The first is to process the documents to a format where it is easier for the classifier to process the information. Two of the most common ways to extract information from a document is to either split the document into terms and store these in an inverted index or to extract keywords from the documents that describes the content. Terms in this case will usually be words and the index will typically be used by statistical methods such as tfidf[4]. For keyword extraction there are some different possibilities and these will be discussed further in section 3.2. The index of term solution is usually good for corpus based solutions where methods such as tfidf[4] weighting is used. The keywords for a document can either be manually assigned or extracted from the document with an algorithm.

After the processing of the documents the text categorization system use a classifier to determine what categories the document is most likely to belong to [5, 18, 3]. To determine this the system use a classifier. A classifier will generally consist of two parts namely training and classification[18]. The purpose of the training is to generate a model from the documents in the training set. This model is used by the classifier to determine the likelihood of a document belonging to a category. The classifier will then select the most probable category and assign the document to this category.

Multi-Label A special case of classification is when a document can belong to more than one category. Multi-label classification has traditionally only been used in text categorization and medical diagnosis [42], but it is increasingly required by other classification tasks. Multi-labeling is more complex than single labeling and some research has been performed to find which combination of methods that give the best results[42].

For the evaluation of this project only single-label classification is used.

Multi-labeling is probably a more realistic scenario for the news article domain, but this introduces more variables that are complex to understand and difficult to control.

3.2 Keyword Extraction

Keyword extraction is to find words that ideally describes the content of a document. When keyword extraction is used in information retrieval the keywords should also describe the document in a way that separates it from the other documents in the collection. Two domains that are almost equal to keyword extraction is keyphrase extraction and key sentence extraction. Keyphrases are parts of sentences that can consist of one to several words. They are used in the same domains as keyword extraction, but can sometimes provide more contextual information and has the ability to capture multi-word expressions. When keyword or keyphrase extraction is used in information retrieval the keywords should also describe the document in a way that separates it from the other documents in the collection. Key sentence extraction extracts entire sentences from a document and is used in slightly different domains. The most common use of it is in automatic summarizing of documents[6]. In this project we will focus on keyphrase extraction, but since keyphrase and keyword extraction are in the same domain we use the terms interchangeably.

3.3 News Articles Domain

News articles on the web have some special traits that will need to be considered when our system is developed. The first of these are that new pages are added many times every day. This means that an

approach that requires the entire collection can be problematic and an approach that can deal with new documents without re-weighting everything would be better for this. The second trait is the language used in news articles which can consist of new and unknown words, citation of persons and mixing of topics. The third trait is the structure of a news article. They usually consist of a title, introductory paragraph and the body text. The title can contain valuable information that represent the article, but it can also be ambiguous or plainly misleading due to the demand for a title that attracts attention from the readers. The introductory paragraph will often summarize the article and is usually less misleading than the title.

3.4 Individual Document Extraction vs. Corpus Based Extraction

Documents are usually classified by using either the information from the document itself or the information from a collection that it belongs to. Using a collection will generally give better results and more importantly it is a lot easier to implement an algorithm that extract statistical data from the entire collection. A typical approach would be to use the tfidf score for the entire collection and choose the words with the highest score as keywords for the documents. There are however some drawbacks with using the collection for classification. One of these is that the entire collection must be collected before the classification or the documents will need to be reclassified when new documents are added to the collection. Compared to individual document extraction the collection based approach will have more problems with scalability and this will influence the processing time.

The field of individual document keyword extraction is relatively new, but some of the difficulties have been identified[19, 2]. The most obvi-

ous problem is to select terms that not only describes the document, but that distinguish it from other documents. This can be difficult when the other documents are unknown, but some optimistic results where reported in [2].

3.5 Term Selection Approaches

Term selection approaches are methods for extracting candidate phrases from a text. In this section some of these methods are described.

3.5.1 N-Grams

N-grams are a sub sequence of items from a sequence. In this work the items used for the n-grams will be words. The n-grams consists of a number of words equal to the n count that occurs in a sequence. N-grams consisting of 1,2 or 3 words are usually referred to as unigram, bigram and trigram, respectively. In keyphrase extraction some additional restrictions are usually added to reduce the number of n-grams from a document. Some of these restrictions is to remove n-grams that starts or ends with a stopword, remove numbers and stemming was applied[8, 9].

3.5.2 Noun Phrase Chunking

Noun phrase chunking or noun phrase extraction, extracts all the noun phrases from a document and use these as potential keywords[2, 8].

3.5.3 Part-of-Speech Tag Patterns

Part-of-speech tag patterns uses patterns of word classes and extract candidate phrases that matches these patterns. An example pattern is: *Adjective Noun*[8].

3.5.4 Stopword Limited Phrases

The text is split into phrases using the stopwords as delimiters between the phrases. An example with “the” as the only stopword would be: *Howard is the new guy*. The resulting phrases would be: *Howard is* and *new guy*. This approach is used in[1].

3.5.5 Graph Based Co-Occurrence

Graph based co-occurrence uses a window to determine if words co-occur. When two words occur within the window size they are said to co-occur. The window size can range from 2 and upwards. A window size of 2 means that words only co-occur when they are adjacent to each other. Words that co-occur are added to the graph as vertices and an edge between the words is created. This approach is used in [6]

3.5.6 Syntactic Filter

A syntactic filter is a filter that only accepts words that belong to a certain word class. This could for instance be only nouns or nouns and verbs. Many of the approaches that uses a syntactic filter have best results with a filter that allows only nouns and adjectives. This approach is used in [6].

Chapter 4

Related Work

In this chapter we start with the work conducted in the document content extraction field. At the end of the chapter methods for individual document keyword extraction, which most closely resemble this work, is presented.

4.1 Machine Learning

Machine learning methods use documents with known keywords to train a classifier. The model generated with this classifier is then used to find keywords in new documents without assigned keywords.

4.1.1 Kea and Kea++

Frank et al.[9] use a N ave Bayes model to extract keyphrases. They first pre-process the text by splitting it up according to phrase boundaries, removing phrases that start with or end with a stopword, per-

form case folding and stemming. A model that uses the tfidf score is built and the distance from the start of the document until the first occurrence of the phrase. The assumption that distance and tfidf are independent is made and they apply Näive Bayes formula to calculate the probability that a phrase is a keyphrase. They use this method to train the algorithm and the resulting model from the test documents is used to extract keywords from new documents in combination with tfidf and distance scores.

Kea++[11] enhances kea with three new features. The first is a controlled vocabulary of terms. The second is that the length of the candidate words is added to the model and the third is the node degree. The node degree is the number of links in the thesaurus that link the words to other candidate terms. A term that has many links to other candidates is more likely to be significant for the document.

4.1.2 Hulth

In her work Hulth[8] uses machine learning to extract keywords. The keywords are extracted from abstracts in the Inspec database. Hulth experiments with three different approaches for selecting terms. These are N-grams, NP chunking and PoS tag patterns. For n-grams uni-grams, bigrams and trigrams were used and those terms that started or ended with a stopword were removed. The chunking is performed with a partial parser. The PoS tagging patterns used 56 patterns and extracted phrases that matched these patterns.

Hulth used four features in her experiment:

1. Within-document frequency
2. Collection frequency

3. Relative position of the first occurrence (the proportion of the document preceding the first occurrence).
4. PoS tag sequence for a term consisting of several tokens

The first 3 were identical to those used by Frank et al.[9] and the fourth merges the PoS tags from the tokens into a new tag of the form $\langle \text{Tag1} \rangle_ \langle \text{Tag2} \rangle$.

The algorithm is trained with positive examples for manually assigned keywords and negative for those that are not manually assigned. It then generates n classifiers with a set of rules from the examples and finally uses these classifiers in a voting scheme to classify an instance.

4.2 Other Approaches

In this section an approach that has some similar features to individual document extraction is described.

Neighborhood Knowledge Wan and Xiao[10], experiments with using knowledge from similar documents in their research. They use a cosine similarity measure to retrieve a k number of similar documents and group these documents in a set. They then create a global word graph from all the documents in the document set, but before the words are added they have to pass a syntactic filter that only accepts nouns and adjectives. They then give each edge a weight that is obtained from two sources. The first is the semantic similarity of the document the words that co-occur in and the original document. The second is the number of co-occurrences in the document. The score of each word is then calculated as the sum of the words that are linked with it multiplied by the weight of the edge that links.

When the candidate words have been computed for the document set they mark the candidate words that appear in the original document. Adjacent phrases are collapsed and only phrases that ends with a noun are accepted. Finally these are ranked and a static number of the top ranked m phrases are extracted as keywords.

4.3 Individual Document Extraction

In this section some algorithms that extract a set of phrases from an individual document are described. The extracted phrases describe the semantic content of the document.

4.3.1 TextRank

The TextRank system[6] uses a graph-based ranking algorithm to extract keywords. It uses many of the ideas that lies behind the well known PageRank[7] algorithm from information retrieval. To rank the vertices in the graph they use a modified version of the original PageRank formula. The original formula was:

$$S(V_i) = (1 - d) + d * \sum_{j \in \text{In}(V_i)} \frac{1}{|\text{Out}(V_j)|} S(V_j)$$

Where V_i is the vertice, $\text{In}(V_i)$ is the set of edges that point to the vertice, $\text{Out}(V_i)$ is the set of edges that the vertex points to and d is a dampening factor that can be set between 0 and 1.

For a web page a link that the edges represent is a seldom thing, but in a document of words the edges between words will be a lot more common. Therefore the TextRank system weights the edges to represent the strength of the connection between two vertices V_i and V_j as a weight w_{ij} . To incorporate the weighting in the original formula they define the score of a vertex to be:

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

To restrict the growth of the graph they only allow single word entries to be added to the graph and merge adjacent keywords to form multi-word keywords in the post processing step. Before a word is added to the graph they need to pass through a syntactic filter. They achieved their best results with a filter that only allowed nouns and adjectives. To identify relations between two lexical units (edges) they use co-occurrence with a window size between 2 and 10 words. When a word pass through the syntactic filter it is added as a vertice to the graph. When a word is co-occurring with another word in the graph an edge is added between these two words in the graph. They use bidirectional and unweighted edges. A graph from an example text in the article[6] is shown in figure 4.1.

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.

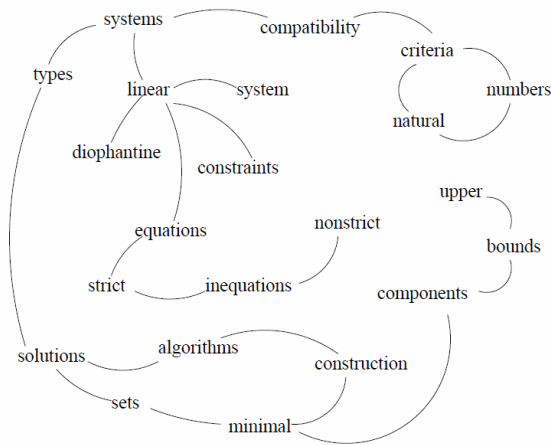


Figure 4.1: TextRank Graph[6]

After the graph has been constructed all vertices are set to an initial value and the algorithm runs in iterations until it the differences V_i and V_{i-1} converges below a threshold.

4.3.2 RAKE

In the RAKE system[1] stopwords are used to split the document into potential keyword phrases. The text between two stopwords is considered to be a phrase. To calculate the best keyword phrases the number of occurrences of each word is counted and a word degree is

calculated. The word degree is the number of words in the phrases where the word occurs added together for all occurrences of the word. So if the word algorithm occurred by itself once and in the phrase: “corresponding algorithms” it would get a word degree of $1 + 2 = 3$.

To calculate the most relevant keywords the word degree divided by the word occurrence is used. The phrases then gets the combined score of the words that occur in the phrase. This approach tends to favor longer phrases and therefore it is important to choose stopwords with care to avoid long phrases with little relevance to be chosen as a keyword phrase. Since the stopwords are so important for this method the article also investigates different methods for choosing stopwords and concludes with a method that chooses the most frequent terms in an initial collection and then removes words that occurs more often in the keywords for the documents than next to the keywords.

4.3.3 Multilingual Single Document Keyword Extraction

In Bracewell et al.[2] processing speed and multilingual support is emphasized. To achieve this they separated the extraction into different phases as illustrated in figure4.2. The first part of their algorithm is morphological analysis and they perform the these four steps during this phase:

1. Word Segmentation
2. Part-of-Speech Tagging
3. Stemming
4. Unigram Frequency Calculation

The word segmentation is to support languages that requires this such as Chinese. Taggers and stemmers are chosen selected from well known alternatives and are language dependent.

For the second part of the algorithm, noun phrase extraction and scoring, noun phrases are extracted and a chunking algorithm is used. After the noun phrases are extracted the frequency of the noun phrases and the individual words in these are counted. Finally the unigram frequency for a noun phrase and the score for a noun phrase is calculated.

In the third phase the noun phrases are clustered. Two phrases are clustered if they have a word in common. Finally the clusters gets a scored by calculating the average noun phrase score of the phrases in the cluster.

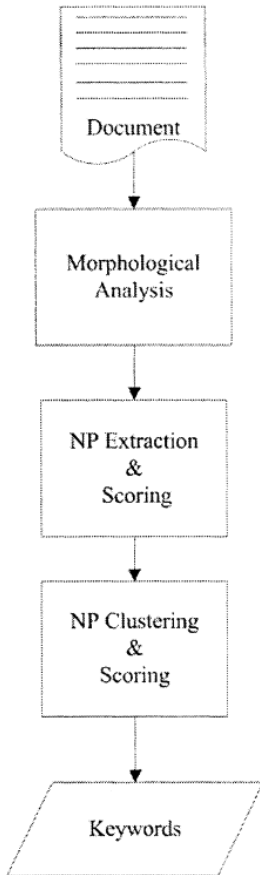


Fig.1. Algorithm Overview

Figure 4.2: Keyword Extraction process[2]

Part III

Prototype Implementation

Chapter 5

Approach

In this chapter we examine the possible methods that we can use for the different phases of the system. We start with an overview of the phases that we want to include in the system and a brief description. This is followed by a discussion of the results achieved by previous methods as well as a comparison between these methods.

5.1 Constraints

The major constraint for this project is that only Norwegian documents are used. This imposes some challenges when it comes to usage of natural language processing tools such as taggers and stopword lists.

5.2 Comparison of Individual Document Keyphrase Extraction Methods

To achieve better results than previous methods in a field of research two basic options exists. One can either attempt to create a new method that might produce better results or attempt to improve an existing method. In this project we work with a specific set of data as described in section 3.3 and we can therefore exploit some of the traits in this data. Using these traits in conjunction with a more general algorithm can improve the results. Therefore we opted for improving an existing method. To find a suitable algorithm we analyzed the individual document keyphrase extraction methods described in chapter 4.

Hulth used n-grams, noun phrase chunking and part-of-speech patterns in her work[8]. She achieved the best results, measured in F-score, by using n-grams in combination with tagging. This might seem a bit counter intuitive since the patterns uses more semantic information than the n-gram approach. This suggest that complex approaches for selecting phrases does not necessarily improve the quality, but the use of a tagger seemed to improve the results substantially.

The TextRank system[6] performed better than the n-gram approach from Hulth on the same test set. The problem with the TextRank system is that it is computationally expensive because of many iterations.

The RAKE system[1] reported slightly better results than the TextRank system, but the computational cost is a magnitude lower. They use the stopword limited phrases method described in section 3.5. The RAKE method is also from 2010 and therefore relatively new compared to the others. This indicates that less experimentation has been perform with the RAKE method. TextRank has been considered by many

to be a state-of-the-art algorithm for individual keyphrase extraction and some classification systems have used TextRank in the classification process[40, 41]. RAKE is therefore considered to be the most advantageous algorithm to serve as a basis for our system. A surprising fact with RAKE is that it uses very few of the standard methods that are present in most of the other algorithms. In this project we will therefore experiment with using some of these methods in conjunction with the RAKE algorithm.

The problem with the RAKE approach is that it tends to favor long phrases. This proved to be a problem for news articles since very few of the selected keywords occurred in several documents. This makes it ill suited for text classification systems and similar domains. To resolve this issue they used a slightly different method that is based on word degree, see section 4.3.2, and tend to favor shorter phrases that occur more often. These shorter phrases usually occur across more documents than the longer phrases and are therefore more ideal for classification systems. In our system we use a slightly different approach to retrieve short phrases that are likely to occur in several documents and we discussed this approach in the next section.

The differences from the RAKE approach for the second phase of our system is the weighting of words based on where in the text they occur and that we use noun phrases. The noun phrases is in the first approach found with a word-bank database[13] that contain all listed Norwegian words and their word class. In the second approach we used a tagger[14]. Noun phrases are used in many approaches and they are more likely to be keyword phrases. A problem with noun phrases is that they require some sort of tagger to find the phrases. Taggers are expensive in computational power and therefore the word-bank approach, which is a simple look up operation, is more desirable if similar results can be obtained.

The results reported by RAKE, TextRank and from Hulth on the same

data set is displayed in figure 5.1.

Method	Extracted keywords		Correct keywords		Precision	Recall	<i>F</i> -measure
	Total	Mean	Total	Mean			
RAKE ($T = 0.33$)							
KA stoplist ($df > 10$)	6052	12.1	2037	4.1	33.7	41.5	37.2
Fox stoplist	7893	15.8	2054	4.2	26	42.2	32.1
TextRank							
Undirected, co-occ. window = 2	6784	13.6	2116	4.2	31.2	43.1	36.2
Undirected, co-occ. window = 3	6715	13.4	1897	3.8	28.2	38.6	32.6
(Hulth 2003)							
Ngram with tag	7815	15.6	1973	3.9	25.2	51.7	33.9
NP chunks with tag	4788	9.6	1421	2.8	29.7	37.2	33
Pattern with tag	7012	14	1523	3	21.7	39.9	28.1

Figure 5.1: RAKE results[1]

5.3 Overall Architecture

The system for document keyword extraction consist of five major components: parser, tagger, find candidate phrases, calculate scores and clustering. These components are depicted in figure 5.2. The input to the system is a set of one or more html documents. The output is a set of phrases that describe the content of the documents.

The parser retrieves a html document from an url address and transform it to plain text. The tagger then tags the words in this text. This is followed by splitting the text into candidate phrases. Scores

are calculated for every phrase and the phrases that have a high degree of similarity are clustered. The top third of the phrases with highest scores are then selected as keyword phrases for the documents.

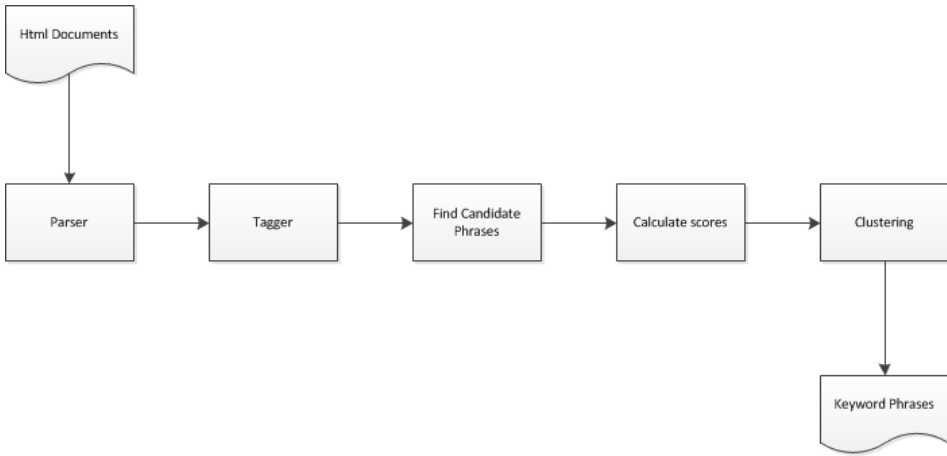


Figure 5.2: Package Diagram

5.4 Stopword List

This system select the candidate phrases based upon a stopwords list. Therefore the stopwords list itself is a vital part of the system and the quality of the stopwords list will propagate throughout the system.

The common approach for most systems is to use a standard list for the given language. For English several standard list exists[33, 34]. For Norwegian it is more difficult to find a stopwords list that is properly tested. The author was unable to find a good stopwords list in any scientific article and instead used the list from [29]. This includes a stopwords list that is a compilation of the lists from several different

sites[30, 31, 32] and in addition some New Norwegian specific stopwords. The list contains 216 stopwords and is therefore a lot smaller than the classic Fox stop list[33] for English which contains 421 words. It was however the largest reliable stop word list that the author could find. To account for this uncertainty the list is compared to an automatically generated stopword list.

The problem is that some domain specific words that carry little informational value will not be present in a general stopword list. A common trait for most stopwords is that they have a high term frequency in the texts and therefore are poor discriminators. Surprisingly enough very little research has been performed in the field of stopword list generation. A naive approach for automatically generating a stopword list is therefore to take the X terms with highest frequency. In [1] they argue that domain specific words that carry a lot of information can also be frequent terms in these domains. To find these words within the frequency list they used they keywords that were labeled to the documents in their training set to determine words that frequently occur within keyword phrases. These words were then removed from the stopword list.

In this system we do not have access to keyword labeled documents and therefore another approach is required. In [28] they attempted to use random sampling and re-weighting of terms, but their results were not very good and their normalized IDF method performed better. A more promising approach is attempted in [35] where several methods are evaluated. Odds Ratio is one of the most promising approaches and it clearly outperforms IDF in their experiments. Therefore this project uses Min(Odds Ratio) to automatically generate a stopword list in addition to the standard list.

Odds Ratio (OR) :

$$OR(t_j, c_k) = \frac{odds(t_j|c_k)}{odds(t_j|^{-}c_k)} = \frac{ad}{bc}$$

where $a = P(t_j, ck)$, $b = P(t_j, \bar{ck})$, $c = P(\bar{t}_j, ck)$, and $d = P(\bar{t}_j, \bar{ck})$.

The words with the lowest scores are the words with the least information and therefore the 300 words with least value is used as a stopword list.

Chapter 6

Implementation

In this chapter the implementation of the three phases of the system are described.

6.1 HTML Parsing

In this section we look at the implementation of the html parsing phase. The details of the implementation is explained and some of the implementation issues and decisions are discussed. This process is depicted in figure 6.1.

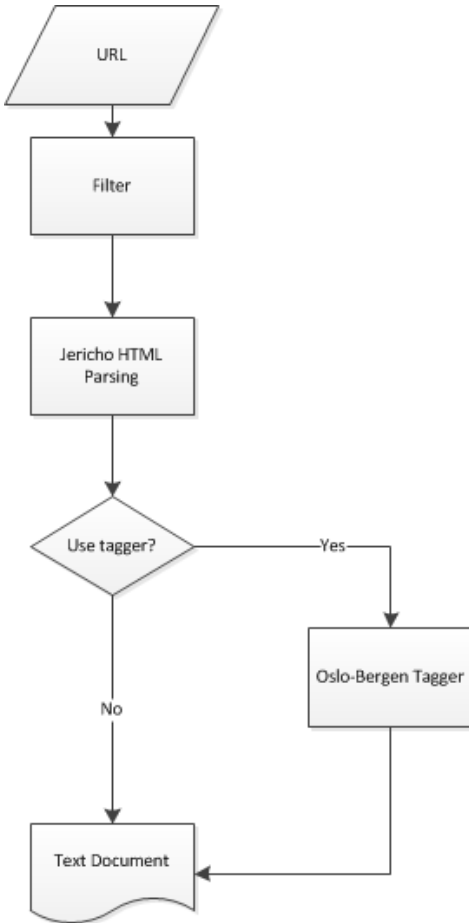


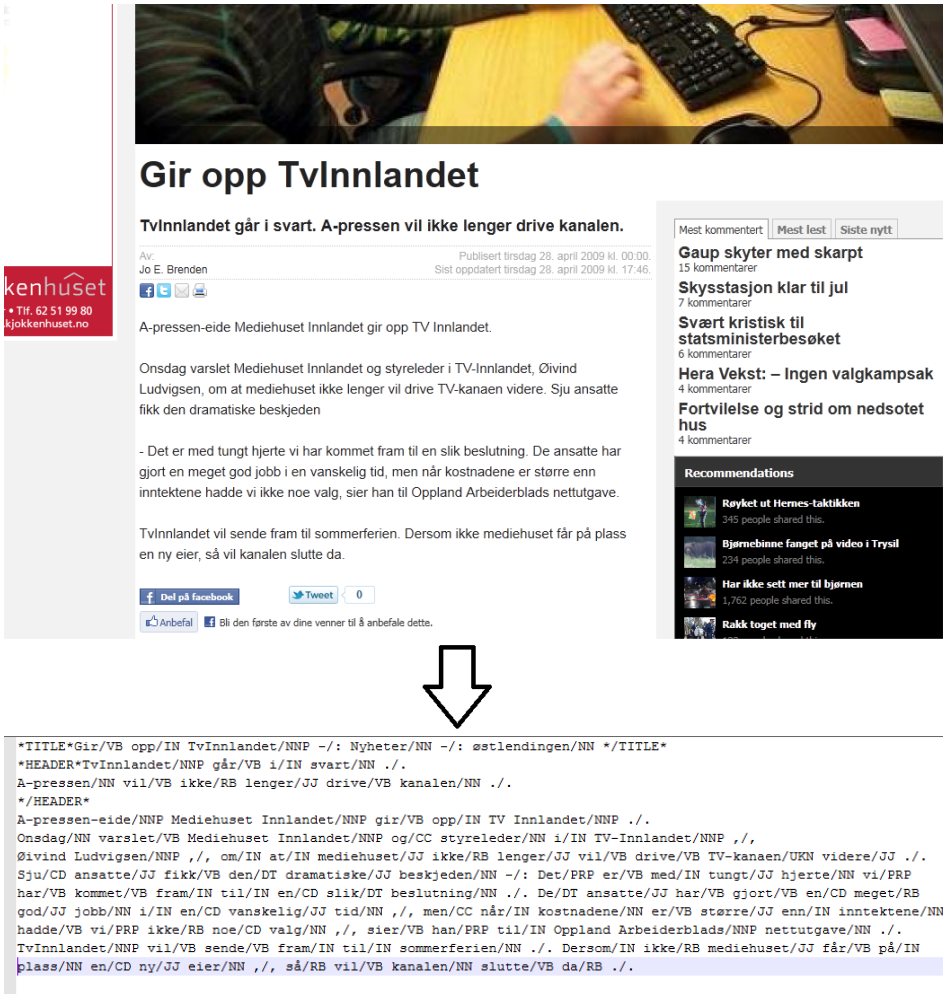
Figure 6.1: HTML Parsing Process

The starting point of the Parsing phase is an url address. From this address we obtain the html page with all the additional adds and links like displayed on the left side of figure 6.2. A filter is then applied to the page to the page to extract the part of the page that marks the start and end of the article text. The text will still contain html tags and

some noise from the social media share icons textual representation. To remove this we use the TextExtractor class from the Jericho HTML Parser[12].

An issue at this point was that the text extractor removes all formatting when it converts the html to plain text. For this system that was a problem since we wanted to give the words occurring in the leading paragraph more weight in the second phase of the system. To maintain the structure of the texts and exploit the news article format described in section 3.3, the texts were split into title, lead paragraph and body text. This allows for weighting based on where words and phrases occur in an article and similar approaches have been attempted before with web documents [16]. The text extractor removes all formatting when it converts the html to plain text and it was modified to allow some formatting tags to remain. These tags were then replaced by the appropriate formatting operation and the title and leading paragraph were respectively marked with *TITLE* and *HEADER* tags as in the right side of figure 6.2.

For the implementation of the system that uses the Norwegian word-bank the parsing phase is completed. For the Oslo-Bergen-Tagger implementation we also employ the tagger at this point. The tagger replaces each word by a word/tag combination. An example is kaste/VB which means that “kaste” is a verb.



Gir opp TvInnlandet

TvInnlandet går i svart. A-pressen vil ikke lenger drive kanalen.

Av: Jo E. Brenden
Publisert tirsdag 28. april 2009 kl. 00:00.
Sist oppdatert tirsdag 28. april 2009 kl. 17:46.

A-pressen-eide Mediehuset Innlandet gir opp TV Innlandet.

Onsdag varslet Mediehuset Innlandet og styreleder i TV-Innlandet, Øivind Ludvigsen, om at mediehuset ikke lenger vil drive TV-kanalen videre. Sju ansatte fikk den dramatiske beskjeden

- Det er med tungt hjerte vi har kommet fram til en slik beslutning. De ansatte har gjort en meget god jobb i en vanskelig tid, men når kostnadene er større enn inntektene hadde vi ikke noe valg, sier han til Oppland Arbeiderblads nettutgave.

TvInnlandet vil sende fram til sommerferien. Dersom ikke mediehuset får på plass en ny eier, så vil kanalen slutte da.

Del på facebook | Tweet | 0

Anbefal | Bli den første av dine venner til å anbefale dette.

Recommendations

- Røyket ut Hernes-taktikken
345 people shared this.
- Bjørnebinne fanget på video i Trysil
234 people shared this.
- Har ikke sett mer til bjørnen
1,762 people shared this.
- Rakk toget med fly

```
*TITLE*Gir/VB opp/IN TvInnlandet/NNP -/: Nyheter/NN -/: østlendingen/NN */TITLE*
*HEADER*TvInnlandet/NNP går/VB i/IN svart/NN ./
A-pressen/NN vil/VB ikke/RB lenger/JJ drive/VB kanalen/NN ./
*/HEADER*
A-pressen-eide/NNP Mediehuset Innlandet/NNP gir/VB opp/IN TV Innlandet/NNP ./
Onsdag/NN varslet/VB Mediehuset Innlandet/NNP og/CC styreleder/NN i/IN TV-Innlandet/NNP ./,
Øivind Ludvigsen/NNP ./, om/IN at/IN mediehuset/JJ ikke/RB lenger/JJ vil/VB drive/VB TV-kanalen/UKN videre/JJ ./
Sju/CD ansatte/JJ fikk/VB den/DT dramatiske/JJ beskjeden/NN -/: Det/PRP er/VB med/IN tungt/JJ hjerte/NN vi/PRP
har/VB kommet/VB fram/IN til/IN en/CD slik/DT beslutning/NN ./ De/DT ansatte/JJ har/VB gjort/VB en/CD meget/RB
god/JJ jobb/NN i/IN en/CD vanskelig/JJ tid/NN ./, men/CC når/IN kostnadene/NN er/VB større/JJ enn/IN inntektene/NN
hadde/VB vi/PRP ikke/RB noe/CD valg/NN ./, sier/VB han/PRP til/IN Oppland Arbeiderblads/NNP nettutgave/NN ./
TvInnlandet/NNP vil/VB sende/VB fram/IN til/IN sommerferien/NN ./ Dersom/IN ikke/RB mediehuset/JJ får/VB på/IN
plass/NN en/CD ny/JJ eier/NN ./, så/RB vil/VB kanalen/NN slutte/VB da/RB ./
```

Figure 6.2: HTML Parsing

6.2 Keyphrase Extraction

The process for this phase is displayed in figure 6.3. The purpose of this phase is to transform plain text into keyphrases with a score value. To find these the first step is to obtain candidate phrases. In our implementation this is performed in the same way as in the RAKE system[1]. We use a stopword list and the text between two stopwords is added as a candidate phrase. We then calculate the values of the different words. Before the phrase score calculation we remove the phrases that does not contain a noun. A value is calculated for each of the remaining phrases and finally the phrases are stemmed using a light stemmer based on the Norwegian Wordbank[13].

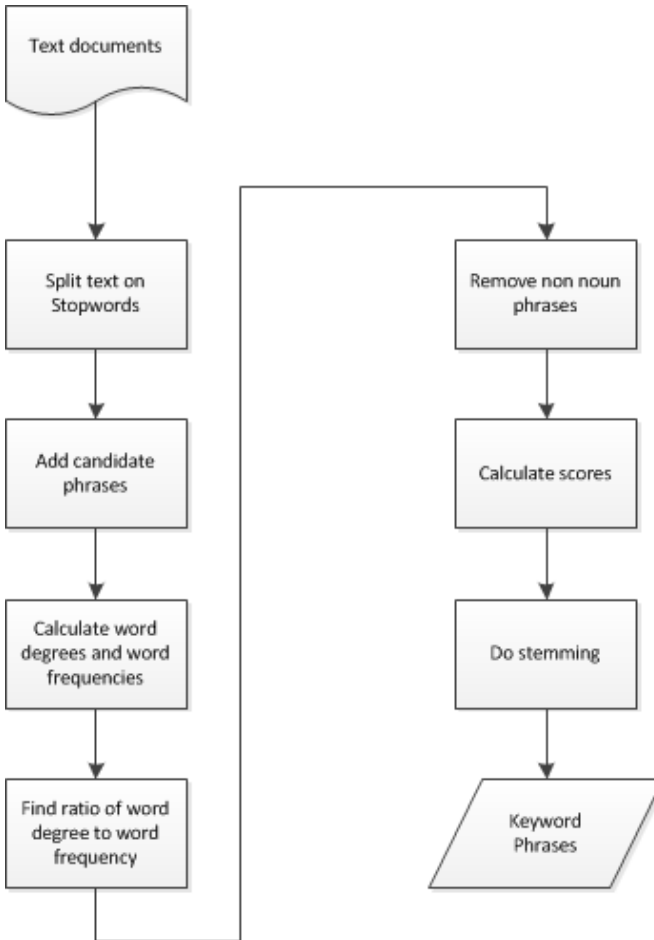


Figure 6.3: Keyword Extraction Process

To illustrate the operations that are performed during this phase we use a news page from the Østlendingen corpus as an example text. The text is displayed in figure 6.4.

Brannsjef Haagenrud i ny jobb - Elverum - Østlendingen Nils Erik Haagenrud (50) slutter som brannsjef i Midt-Hedmark for å ta samme jobb i Fredrikstad. – Jeg vil søke nye utfordringer. Fredrikstad er en by med 74.000 innbyggere og har et langt større brannvesen. Her vil det være nye organisasjonsmuligheter, og en spennende og interessant arbeidsplass, sier Haagenrud. Pendler I første omgang skal han ha ett års permisjon fra jobben som daglig leder i Midt-Hedmark brann- og redningsvesen IKS. Haagenrud skal pendle mellom Osen og Fredrikstad. Trives I januar i år gikk det en reklamekampanje hvor Haagenrud som brannsjef anbefalte sikkerhetselskapet G4S. Han fikk kritikk for å ha gått ut over rollen sin og hans habilitet ble diskutert. – Hvor mye har denne hendelsen hatt å si for valget om ny jobb? – Nei, ingenting. Jeg trives og har det veldig bra i min jobb nå, sier Haagenrud. Han har vært brannsjef siden 2001, og jobbet i brannvesenet i Elverum siden 1988. Ifølge Haagenrud skal det konstitueres en brannsjef det året han er permittert. Etter det Østlendingen erfarer vil Terje Hansen fra Tynset/Tolga vikariere for Haagenrud.

Figure 6.4: Sample Document Text

6.2.1 Candidate Extraction

The first operation is the candidate extraction. The text between two stopwords is considered to be a candidate phrase and is added to the candidate list. We also consider numbers and words without letters to be stopwords. The results of applying this step to the text in figure 6.4 can be seen in figure 6.5.

brannsjef haagenrud - ny jobb - nils erik haagenrud - slutter -
 brannsjef - midt-hedmark - ta samme jobb - fredrikstad - søke -
 utfordringer - fredrikstad - by - innbyggere - langt større brannvesen
 - organisasjonsmuligheter - spennende - interessant arbeidsplass -
 haagenrud - pendler - første omgang - ett års permisjon - jobben -
 daglig leder - midt-hedmark brann - redningsvesen iks - haagenrud -
 pendle mellom osen - fredrikstad - trives - januar - gikk -
 reklamekampanje hvor haagenrud - brannsjef anbefalte
 sikkerhetselskapet g4s - kritikk - gått - rollen - hans habilitet -
 diskutert - hvor - hendelsen hatt - si - valget - ny jobb - nei -
 ingenting - trives - veldig bra - min jobb nå - haagenrud - brannsjef
 siden - jobbet - brannvesenet - siden - ifølge haagenrud -
 konstitueres - brannsjef - året - permittert - erfarer - terje hansen -
 tynset/tolga vikariere - haagenrud

Figure 6.5: Candidates

6.2.2 Wordscore Calculation

To calculate the value of every word in the document we use three metrics. Wordfrequency is the number of times a word appears in the text. Worddegree is the sum of the length of the phrases that the word occurs in. Word ratio is worddegree divided by the wordfrequency.

The wordscore calculation process can be demonstrated with the text contained in figure 6.4. The text is first split into tokens, which in the case of the word-bank approach will mean single words. With the tagger approach it will usually also be words, but the name of an entity can sometimes consist of more than one word and will therefore consist of several words. A typical example is the name of a person where the token will consist of two or more words.

After the text has been split up the frequency of every word is counted

and the results are stored in a list. The system then processes the words in this list and determines the worddegree by checking the number and length of the candidate phrases that the words occur in. An example is the word “siden” that appears in the two phrases “siden” and “brannsjef siden”. The length of the first phrase is one and the length of the second is two so the worddegree for “siden” becomes three.

The word ratio is worddegree divided by wordfrequency and the word “siden” will get a word ratio of $3/2 = 1,5$. This is the value that we use at later stages of the phase when we calculate the phrase scores.

6.2.3 Noun Phrase Filter

Very few phrases that does not contain a noun will be good keyword phrases and therefore we decided to apply a noun phrase filter. The filter checks the words in the phrase and if none of the are a noun the phrase will be removed. The noun phrase filter method has also been used in several other articles before[2, 15]. The word bank approach check if the noun list contains the word and in the tagger approach the tag of the word is checked.

6.2.4 Phrase Score Calculation

The final part of the Keyphrase Extraction phase is the calculation of the phrase scores. This is a simple process of adding together the values of the words that the phrase consist of. After the phrase scores have been calculated they are sorted by value and the result from the text in figure 6.4 can be seen in figure 6.6.

6.2.5 Weighting

A news article consist of different parts as described in section 3.3. Not all of these part will be equally relevant for describing the article. To accommodate for this the system weights the different parts of the article differently. The article is split into title, leading paragraph and header. These are all given different weights and we decided to let the leading paragraph receive most weight since the title can sometimes be ambiguous as discussed in section3.3. To illustrate the difference in results between weighting and not weighting we have included both top 10 keyphrase results for the sample text. In figure 6.6 the top 10 results without weighting is displayed. In figure 6.7 the results with weighting is displayed.

The results are in the authors judgment better after the weighting. The top four phrases contain information about job and the person the article is about. These are all relevant phrases. Before the weighting the phrases contain information about “the fire chief recommended a company”, “a one year leave”, “a lot larger fire department” and “commute between Osen”. From these phrase the first is somewhat relevant and the third is a relevant phrase.

An interesting thing to note is that the phrase “ny jobb” occurs twice. This is because the phrase occurs twice in the text and we resolve the issue by clustering phrases in the final phase of the system. “nils erik haagenrud” is also captured as a phrase although it only contains one proper noun. This is a problem that only occurs when the word-bank is used since some words have multiple possible word classes and if any of these are a normal noun the phrase will be considered as a noun phrase.

brannsjef anbefalte sikkerhetsselskapet g4s 14.0, ett års permisjon 9.0, langt større brannvesen 9.0, pendle mellom osen 9.0, min jobb nå 8.5, ta samme jobb 8.5, nils erik haagenrud 7.75, reklamekampanje hvor haagenrud 6.75, ny jobb 4.5, ny jobb 4.5

Figure 6.6: Keyphrase scores top 10

ta samme jobb 46.25, nils erik haagenrud 41.625, ny jobb 20.25, ny jobb 20.25, brannsjef anbefalte sikkerhetsselskapet g4s 18.2, min jobb nå 16.25, brannsjef haagenrud 11.825, reklamekampanje hvor haagenrud 10.625, pendle mellom osen 9.0, langt større brannvesen 9.0

Figure 6.7: Keyphrases top 10 with weighting

6.2.6 Stopword Trigrams

A problem with using stopwords as delimiters for phrases is that phrases that contain stopwords will never be discovered. An example of a phrase that contains a stopword is “axis of evil”. To include phrases that include stopwords as candidates we decided to add stopword trigrams that occurs at least twice in the text as candidates. A stopword trigram consists of the word before the stopword, the stopword itself and the word after the stopword.

6.3 Clustering

The final phase of the system is a set of methods to cluster the keyphrases. The purpose of the clustering is to generate shorter phrases

that occur in more documents and is therefore more relevant for classifiers and similar applications.

The first method is taken from Bracewell et al.[2] and consists of clustering two phrases together if one is a sub-phrase of the other. Unlike Bracewell the clusters get a score equal to the sum of the phrases they contain. The purpose of this is to increase the chances for a concept that is explained in slightly different manners to get a score equal to how relevant it is for the document. The alternative could be several words that fall below the threshold even if they represent the most important concept in the text. Finally we set the name of the cluster to the phrase that all of the phrases contained and the score is set to the sum of all the phrases in the cluster.

Part IV

Evaluation and Conclusion

Chapter 7

Evaluation

In this chapter we evaluate the keyword extraction algorithm by using the extracted keywords with a classifier. The results are compared to another state of the art algorithm individual keyword extraction algorithm.

7.1 The Evaluation Process

To evaluate the results from the keyword extraction algorithm a classifier is used. The process for the evaluation is shown in figure 7.1. The input to the classifier is the keyword phrases that was extracted with the algorithm and the output is the number of correctly classified documents from a test set.

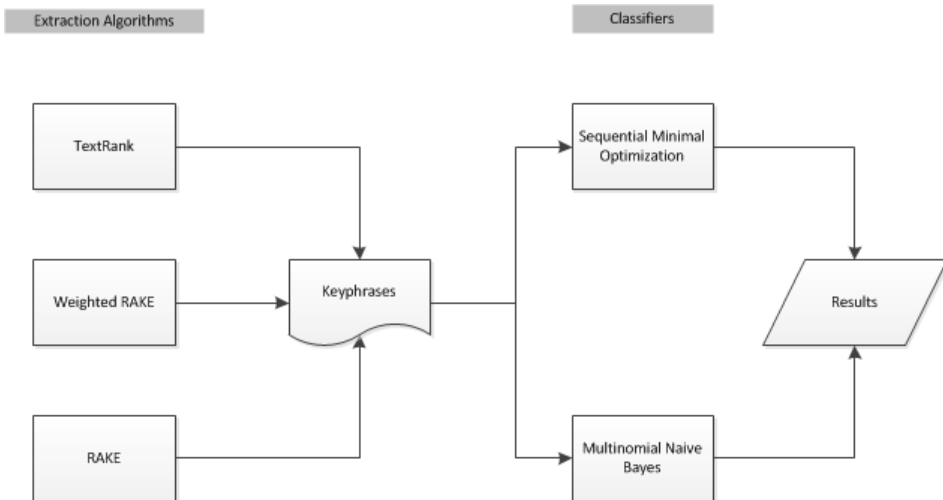


Figure 7.1: Evaluation Process

The keyword extraction algorithms that were used in the evaluation are displayed in figure 7.1.

The TextRank algorithm is described in section 4.3.1. The algorithm has been implemented as described in section 3 of Mihalcea and Tarau[6] and section 4.3.1 in this report with a window size of 2. The difference is that the documents used in this project are in Norwegian and therefore we used the Oslo-Bergen tagger. TextRank has been used in several classification systems and has obtained good results.

Our approach is the system that was described in the chapter 5 and 6.

RAKE is the basic algorithm described in Rose et al.[1].

7.2 The Classification Algorithms

In this section the two classification algorithms are described. The classifiers are displayed in figure 7.1. We decided to use the Multinomial Naive Bayes[24, 22] and sequential minimal optimization (SMO)[21] algorithms to evaluate the results.

The Multinomial Naive Bayes(MNB) is a classic algorithm that has obtain good results in several fields and also in text categorization. The multinomial version of Naive Bayes usually produces good results for text categorization and in general outperforms the multi-variate Bernoulli version[44]. The SMO algorithm trains a support vector classifier. Currently the support vector classifiers are considered to produce the best results[18, 27]. In some rare cases they can however be surpassed by Naive Bayes classifiers and Naive Bayes will also generally require less training than a SVM algorithm[27].

7.3 The Corpus

The corpus that we used during the development of this system was a set of 1100 news articles from the news site ostlendingen.no. For the evaluation of the system this is a rather small set and the statistical uncertainty would be too large to get definitive conclusions. To address this issue a set of 15000 news articles divided into 448 different categories from adresseavisen[25] were used. Each article contains zero or more tags that define the category they belong to. An article that does not have a tag is treated as an unclassified document and therefore not used in the evaluation. These news articles are also categorized by the publisher and gives the classification less bias than if we categorized ourselves. The problem with this approach is that we

rely on the classification performed by others and there is a degree of uncertainty in the quality of the classification.

Single-label classification was chosen as the method for evaluating the system. A problem with this corpus for single label classification is that many documents occur in several categories. To resolve this issue a set of 50 categories where no document occurred in multiple categories was used. All of the categories were selected randomly to remove the potential bias that could occur with manual selection. All of the categories contained at least 5 documents and the total number of documents was 1592.

7.4 Methods

Many of the features that was added to the original RAKE algorithm in this project was added because of a theory that they would improve the results. To measure if these features actually improved the results some parameters that controlled if these features should be included for a test were added. The following is a list of the features that was tested to see if they improved the results:

- **Weighting:**
Weighting of the words based upon where in the text they occur as described in section 6.2.5.
- **Only noun phrases:**
Whether to only used phrases that contained a noun or not as described in section 6.2.3.
- **Clustering**

Clustering of the scored candidate phrases as described in section 6.3.

- Including stopword trigrams
Including trigrams that occurred twice and contained a stopword as described in section 6.2.6.
- Stemming
Stemming of the scored candidate phrases as described in section 6.2.
- Use of ratio or worddegree
Whether to use the ratio of worddegree to wordfrequency or worddegree to score the words in the text.
- Removal of proper nouns
Removing proper nouns before candidate phrases are selected.

All different combinations of these variables were tested to find what methods that improved the results. In addition we also tested the two stopword lists. The standard version of RAKE as it is described in Rose et al.[1] was also tested. Only the SMO classifier was tested for all combinations, but the presented results in the next section was tested with both of the classifiers.

7.5 Results

The results were generated using cross-validation with 10 runs and 5-folds and can be seen in table 7.1. Column 2-5 display the precision achieved with the different classification algorithms and the stopword

lists. SMO is sequential minimal optimization, MNB is multinomial naive bayes, AGS is automatically generated stopword list and NS is the normal stopword list from[29].

Algorithm	SMO and AGS	SMO and NS	MNB and AGS	MNB and NS
TextRank	31.19%	31.19%	38.04%	38.04%
Default RAKE, worddegree	36.16%	34.66%	47.58%	45.05%
Default RAKE, wordratio	36.66%	37.04%	46.40%	45.58%
Weighted RAKE, default	25.45%	25.11%	27.66%	26.97%
Weighted RAKE, best	37.31%	35.67%	47.54%	42.24%

Table 7.1: Results

The algorithm that achieved the lowest score was surprisingly the default weighted RAKE method. The primary reason for the poor performance was the clustering method. In any combination where the clustering was used the precision dropped by approximately 10%. As expected from the results reported from Rose et al.[1] RAKE performed better than TextRank. They also found that using worddegree for categorization of news articles would perform better than using ratio of worddegree to wordfrequency. These results also gave slightly better performance for the worddegree method. The weighted RAKE options that performed best was the weighted RAKE method with the following settings:

- Weighting: yes

- Only noun phrases: no
- Clustering: no
- Including stopword trigrams: yes
- Stemming: yes
- Use of ratio or worddegree: worddegree
- Removal of proper nouns: no

The weighted RAKE method performed about equally good as the default RAKE methods. In these tests we used only two settings for the weights; therefore, optimized weights might yield better performance. However, the indication from these tests are that no significant improvement is obtained by using weights. Still, one of the clear indications is that RAKE outperforms TextRank by a significant amount in all of the tests.

The option that decreased the performance significantly in all cases was, as previously mention, clustering. The stopword trigrams had a very small but positive impact on the results in most cases. Surprisingly only allowing noun phrases decreased the performance. Stemming seemed to be relatively neutral in the effects of the classification.

The two stopword lists were also compared to each other and in general the automatically generated stopword list performed a little better.

Because of the negative results for the use of noun phrases, the Wordbank approach was not tested since it will give exactly the same results as the tagger approach without noun phrases. The only potential difference would be that the tagger is able to recognize multi-words which typically is proper nouns.

Chapter 8

Conclusion

This project has analyzed the current methods for extracting keyphrases from individual documents. We then attempted to create a more specialized algorithm for the news article domain from one of these general algorithms and chose to use RAKE as the general algorithm. The most important part of the new algorithm was the weighting of the words based upon where in the text they occurred and therefore we called it weighted RAKE. The weighted RAKE algorithm and the components were thoroughly analyzed and compared to the RAKE algorithm and TextRank algorithm. The evaluation consisted of classifying the documents from the Adressa news corpus into the correct categories. The evaluation also compared the results from two different stopwords lists and one of these was automatically generated.

The results from the evaluation was somewhat disappointing and did not indicate that weighting improved the baseline algorithm. However there still exists many possible combinations of weighting and other options that could potentially change this. The results does however give a clear indication that the RAKE algorithm performs better than TextRank in this classification task.

An unexpected and positive result was the performance of the automatically generated stopword list. It outperformed the manual standard list by approximately 1-2% in all the tests and this indicates that automatically generated stopword lists could potentially be an asset to natural language processing although more testing is needed.

Chapter 9

Future Work

In this chapter the possibilities for future work with this system are discussed.

9.1 Multi-label Classification

In this project we evaluated the performance of the algorithm with a single-label classifier. For text-categorization systems a common task will be to classify documents that can have multiple labels and therefore a study of the performance for multi-label classification is a potential future task.

9.2 Using Keyphrases to Enhance Existing Systems

Hulth and Megyesi developed a classification system[43] where she was able to improve the results of an existing system by enhancing it with her methods for keyphrase extraction as described in[8]. TextRank has also been used for classification systems[41]. Therefore an interesting project would be to investigate if weighted RAKE could be used to improve the performance for a similar text categorization system.

9.3 Classification System with Semantic Relatedness

An advantage with the individual document keyphrase extraction systems is that they maintain much of the semantic context from the documents. Most classifiers can only work with exact matches between attributes and some of the semantic context that exists in phrases are therefore lost. To use this semantic context a system that measured the semantic relatedness between the phrases in a document and the phrases for a category could be used.

Bibliography

- [1] Rose, S., Engel, D., Cramer, N. & Cowley, W.(2010). Automatic keyword extraction from individual documents. *In: Berry, M.W., Kogan, J (eds.): Text Mining: Application and Theory. John Wiley & Sons, Ltd.*
- [2] Bracewell, DB., Ren, F. & Kuriowa S.(2005). Multilingual single document keyword extraction for information retrieval. *In Proceedings of NLP-KE'05.*
- [3] Fabrizio, S., (2002) Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, Vol. 34, (1).
- [4] <http://en.wikipedia.org/wiki/Tf%E2%80%93idf>, [Downloaded 25 March 2011]
- [5] Bracewell, DB., Ren, F., Kuriowa, S. & Yan J. (2009) Category Classification and Topic Discovery of Japanese and English News Articles. *Theoretical Computer Science* 225 (2009) 51–65
- [6] Mihalcea, R. & Tarau, P. (2004) TextRank: Bringing order into texts. *In Proceedings of EMNLP 2004* (ed. Lin D and Wu D), 404–411.

-
- [7] Brin, S. & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7).
- [8] Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, Japan, August.
- [9] Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C. & Nevill-Manning, C. G. (1999). Domain-specific keyphrase extraction. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 668–673, Stockholm, Sweden.
- [10] Wan, X. & Xiao, J. (2008). Single Document Keyphrase Extraction Using Neighborhood Knowledge. In *Proceedings of AAAI*, (2008), 855-860.
- [11] Medelyan, O. & Witten, I. H. (2006) Thesaurus based automatic keyphrase indexing. In *Proceedings of the Joint Conference on Digital Libraries* (2006), 296-297.
- [12] <http://jericho.htmlparser.net/docs/index.html>, [Downloaded 7 April 2011]
- [13] <http://www.edd.uio.no/prosjekt/ordbanken/index.html>, [Downloaded 7 April 2011]
- [14] <http://tekstlab.uio.no/obt-ny/index.html>, [Downloaded 7 April 2011]
- [15] Barker, K., & Cornacchia, N. (2000). Using nounphrase heads to extract document keyphrases. In *Canadian Conference on AI*.

- [16] Desmontils, E. & Jacquin, C. (2002). Indexing a Web site with a terminology oriented ontology. *In: Cruz IF, Decker S, Euzenat J, McGuinness DL (eds) The emerging semanticWeb. IOS Press, Amsterdam, 181–198.*
- [17] <http://www.cs.waikato.ac.nz/ml/weka/>, [Downloaded 29 April 2011]
- [18] Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. *In Proceedings of ECML-98, 10th European Conference on Machine Learning (Chemnitz, DE, 1998), 137–142.*
- [19] Turney, P. D. (2000) Learning Algorithms for Keyphrase Extraction. *Information Retrieval (2)*, 303-336.
- [20] http://www.cis.hut.fi/research/som_pak/, [Downloaded 12 May 2011]
- [21] Platt, J. (1998). Fast Training of Support Vector Machines using Sequential Minimal Optimization. *Advances in Kernel Methods - Support Vector Learning, B. Schoelkopf, C. Burges, and A. Smola, eds., MIT Press.*
- [22] John, G.H. & Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence.* 338-345.
- [23] Quinlan, J. R. (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers.
- [24] http://en.wikipedia.org/wiki/Naive_Bayes_classifier, [Downloaded 15 May 2011]
- [25] www.adressa.no/, visited 15.05.2011

-
- [26] http://en.wikipedia.org/wiki/Support_vector_machine, [Downloaded 15 May 2011]
- [27] <http://nlp.stanford.edu/IR-book/pdf/13bayes.pdf>, [Downloaded 15 May 2011]
- [28] Lo, R. T.-W., He, B. & Ounis, I. (2005) Automatically building a stopwords list for an information retrieval system, *Journal of Digital Information Management*, 3 (1), 3–8.
- [29] <http://www.wis.no/999/147/33899-170.html>, [Downloaded 19 May 2011]
- [30] <http://www.ranks.nl/stopwords/norwegian.html>, [Downloaded 20 May 2011]
- [31] <http://snowball.tartarus.org/algorithms/norwegian/stop.txt>, [Downloaded 20 May 2011]
- [32] <http://dnnspeedblog.com/SpeedBlog/PostID/3180/Norwegian-Stop-words>, [Downloaded 20 May 2011]
- [33] Fox, C. (1990). A stop list for general text. *ACM-SIGIR Forum*, (24), 19-35.
- [34] Van Rijsbergen, C.J.(1979) *Information Retrieval*, (2)., Dept. of Computer Science, University of Glasgow
- [35] Makrehchi, M. & Kamel, M. (2008) Automatic Extraction of Domain-Specific Stopwords from Labeled Documents, *Springer Berlin / Heidelberg*[online], [Downloaded 21 May 2011]
- [36] Phillips, E.M. & Pugh, D.S. (2005). *How to Get a PhD: A Handbook For Students And Their Supervisors*, Open University Press, Buckinghamshire.

- [37] Hevner, A. R., March, S. T., Park, J. & Ram, S. (2004). Design science in information systems research, *MIS Quarterly*, 28(1),75–105.
- [38] Klein, H. K. & Myers, M. D. (1999). A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems, *MIS Quarterly*,23(1), 67-93.
- [39] Lee, A. S., Baskerville, R. L., Liebenau, J., & Myers, M. D. (1995) Judging Qualitative Research in Information Systems: Criteria for Accepting and Rejecting Manuscripts, *Proceedings of the Sixteenth International Conference on Information Systems*.
- [40] Mihalcea, R. & Hassan, S. (2005). Using the essence of texts to improve document classification. *In Proceedings of the Conference on Recent Advances in Natural Language Processing*.
- [41] Hassan, S. & Banea, C. (2006) Random-walk term weighting for improved text classification, *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*.
- [42] Tsoumakas, G. & Katakis, I. (2007) Multi-label classification: An overview. *International Journal of Data Warehousing and Mining* 3.
- [43] Hulth, A. & Megyesi, B. B. (2006) A study on automatically extracted keywords in text categorization, *Proceedings of the 21st International Conference on Computational Linguistics*
- [44] A. McCallum & K. Nigam. (1998) A comparison of event models for naive bayes text classification. *In AAAI-98 Workshop on Learning for Text Categorization*.

Appendix

Appendix A

Stopword List

A.1 List Generated with Odds Ratio

This is a stopword list that consist of the 300 words with the lowest score using Odds ratio on the adressa corpus.

dagens synes fleste gjelder holdt mål usa rekke finne foran situasjonen dagen spesielt årene gitt begge tiden uke kommet regjeringen skjedde klokken plass kjent natt vant barn bør forhold lite vanskelig vår dersom funnet president følge egen senere lenge kanskje frem ganger byen hver større utenfor land hvert gode morgen nesten hos bra imidlertid jo annen nytt ntb-afp la aldri vei gjorde mindre minst gått mulig stort viktig seks vet millioner politidistrikt dager heller største gamle rett mennesker ett mann klart neste mannen hvordan ti likevel ham beste hvis hatt også stedet gi fordi bedre gir dermed si sett videre ønsker leder adressa.no del torsdag allerede bak tillegg satt derfor måtte tror god meg ligger svært først disse løpet norsk søndag tar gjort samtidig forteller mest veldig prosent du tirsdag kveld se onsdag gjøre står langt sitt fem gjennom like tilbake hans kroner lørdag ville komme litt

trondheim personer grunn fram ny mandag uten stor alt saken nok folk viser les ser oss tok fjor fortsatt godt fredag gjør ned hva skriver opplyser landet igjen tid samme fire sine tatt man gå skulle annet slik norske nye sammen her gang helt kunne dem gikk store norge ta fått hele hvor mye rundt kom politiet mellom ingen tidligere dag kommer både blitt går blant får hun mener når mens sa første denne selv bare siste siden tre mer oslo der mange enn under alle må eller sin bli noe noen fikk få vært inn blir før ifølge opp jeg andre være hadde dette flere så nå to år ntb over mot ha ut ved kan da vil skal også vi han seg etter var men sier et ble fra ikke om den de har med at å for av som er en det på til og i

A.2 Stopword Pre-Defined List

This is the list with 216 words that was found in [29].

alle andre at av bare begge ble bli blir blitt bort bra bruke både da de deg dem den denne der dere deres det dette din disse dit ditt du eller ei en ene eneste enhver enn er et ett etter for fordi forsøke fra fram før først få gjorde gjøre god gå ha hadde han hans har hennar henne hennes her hit hun hva hvem hver hvilke hvilken hvis hvor hvordan hvorfor i ikke ingen inn innen inni ja jeg kan kom kun kunne lage lang lik like man mange med meg meget mellom men mens mer mest min min mitt mot mye må måte ned nei noe noen ny nå når og også om opp oss over på rett riktig samme seg selv si siden sin sine sist sitt sjøl skal skulle slik slutt som start stille så sånn tid til tilbake under ut uten var ved verdi vi vil ville vite være vært vår å blei bae dei deim deira deires di dykk dykkar då eg ein eit eitt elles hjå ho hoe honom hoss hossen ikkje ingi inkje korleis korso kva kvar kvarhelst kven kvi kvifor me medan mi mine mykje no noka noko nokon nokor nokre si sia sidan so somme somt um upp vart varte vere verte vore vors vort

Appendix B

Groups Used in Evaluation

The following is a list of the 50 categories that were used in the evaluation of the algorithms:

Arne, Arve, Aure, Bjarne, Bydrift, Christian, City, Dag, Digital, Dronning, Elv, Folkemusikk, Fosnes, Glatte, Helse, Hurtigruta, Høylandet, Jubileum, Knut, Leka, Levanger, Liv, Midtre, Munkegata, Munkvoll, Nidar, Nidaros, Nidarvoll, Påkjørsel, Reiseliv, Rita, Ritt, Roar, Saupstad, Skansen, Skiskyting, Sluppen St, St. Olavs, Statens Sykepleiere, Sør-Trøndelag, Terror, Trafikkulykke, Turn, Utsteder, Vektene, Voldtekt, Yahoo, Ørland

Appendix C

External Resources

In this section some of the external resources that are used in the system are described.

C.1 Jericho HTML Parser

Jericho HTML Parser[12], is an open source java library that allows the user to analyze and manipulate the content of a HTML document. In this project the text extractor class is used to extract plain text from a html page. The source code for this class was modified slightly to allow certain tags to remain after the text extraction. These tags were then processed in the system to give the plain text the original format. With this approach it is easy to identify line breaks and headings within the text.

C.2 Oslo-Bergen-Tagger

The Oslo-Bergen-Tagger[14], is a morphologic and syntactic tagger for Norwegian. It was developed by the university of Oslo and the company Uni Computing from Bergen.

C.3 Norsk Ordbank

Norsk ordbank[13] (Norwegian wordbank), is a database that contains all lexical base forms connected to the inflected forms of every unit. In this project we use the full word forms in this database. The words are also marked with their word class. This word class field is used by the system when it determines if a phrase contains a noun or not.