DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS

STOCKHOLM, SWEDEN 2020

Keyword Extraction from Swedish Court Documents

SANDRA GROSZ

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

KTH Computer Science
and Communication

Keyword Extraction from Swedish Court Documents

SANDRA GROSZ

Master's Thesis at EECS
Supervisor: Olov Engwall
Examiner: Viggo Kann

February 26, 2020

Abstract

This thesis addresses the problem of extracting keywords which represent the rulings and and grounds for the rulings in Swedish court documents. The problem of identifying the candidate keywords was divided into two steps; first preprocessing the documents and second extracting keywords using a keyword extraction algorithm on the preprocessed documents.

The preprocessing methods used in conjunction with the keywords extraction algorithms were that of using stop words and a stemmer. Then, three different approaches for extracting keywords were used; one statistic approach, one machine learning approach and lastly one graph-based approach.

The three different approaches used to extract keywords were then evaluated to measure the quality of the keywords and the rejection rate of keywords which were not of a high enough quality. Out of the three approaches implemented and evaluated the results indicated that the graph-based approach showed the most promise. However, the results also showed that neither of the three approaches had a high enough accuracy to be used without human supervision.

Keywords: Keywords extraction, Information Retrieval, Natural Language Processing.

Referat

Extraktion av nyckelord från svenska rättsdokument

Detta examensarbete behandlar problemet om att extrahera nyckelord som representerar domslut och domskäl ur svenska rättsdokument. Problemet med att identifiera möjliga nyckelord delades upp i två steg; det första steget är att använda förbehandlingsmetoder och det andra steget att extrahera nyckelord genom att använda en algoritm för nyckelordsextraktion.

Förbehandlingsmetoderna som användes tillsammans med nyckelordsextraktionsalgoritmerna var stoppord samt avstammare. Sedan användes tre olika metoder för att extrahera nyckelord; en statistisk, en maskininlärningsbaserad och slutligen en grafbaserad.

De tre metoderna för att extrahera nyckelord blev sedan evaluerade för att kunna mäta kvaliteten på nyckelorden samt i vilken grad nyckelord som inte var av tillräckligt hög kvalitet förkastades. Av de tre implementerade och evaluerade tillvägagångssätten visade den grafbaserade metoden mest lovande resultat. Däremot visade resultaten även att ingen av de tre metoderna hade en tillräckligt hög riktighet för att kunna användas utan mänsklig övervakning.

Nyckelord: nyckelordsextraktion, informationssökning, naturligt språkbehandling.

Contents

1 Introduction **1**
- 1.1 Problem Statement . . . 1
- 1.2 Purpose . . . 2
- 1.3 Objective . . . 2
- 1.4 Delimitations . . . 2
- 1.5 Ethical, Social and Sustainability Concerns . . . 2

2 Theory **5**
- 2.1 Natural Language Processing . . . 5
  - 2.1.1 Information Retrieval . . . 5
  - 2.1.2 Automatic Summarization . . . 6
- 2.2 Keyword Extraction . . . 7
  - 2.2.1 Statistical-Based Approaches . . . 7
  - 2.2.2 Machine Learning-Based Approaches . . . 9
  - 2.2.3 Graph-Based Approaches . . . 12
- 2.3 Preprocessing Methods . . . 13
  - 2.3.1 Part-of-Speech Tagging . . . 13
  - 2.3.2 Noun Phrase Chunking . . . 14
  - 2.3.3 Stemming . . . 14
  - 2.3.4 Lemmatization . . . 14
  - 2.3.5 Co-occurrence . . . 15
  - 2.3.6 Stop Words . . . 15
- 2.4 Conclusions of the Literature Study . . . 15

3 Method **17**
- 3.1 Architecture . . . 17
- 3.2 Data Set . . . 18
- 3.3 Lexical Analysis . . . 19
  - 3.3.1 Word Tokenization . . . 19
  - 3.3.2 Sentence Tokenization . . . 20
- 3.4 Preprocessing . . . 20
  - 3.4.1 Stop Words . . . 20
  - 3.4.2 Stemming . . . 21

3.5 Keyword Extraction Techniques . . . 23
3.5.1 Statistical-Based Approach: TF-IDF . . . 23
3.5.2 Machine Learning-Based Approach: K-medoid . . . 23
3.5.3 Graph-Based Approach: KeyGraph . . . 24

4 Evaluation **27**
4.1 Subjects . . . 27
4.2 Experimental Documents . . . 27
4.3 The Design of the Evaluation . . . 28

5 Results **31**
5.1 Generated Keywords . . . 31
5.2 Average and Median Score . . . 32
5.3 The Average Deviation . . . 35

6 Discussion **37**
6.1 Statistical-Based Approach . . . 37
6.2 Machine Learning-Based Approach . . . 38
6.3 Graph-Based Approach . . . 38
6.4 Comparison . . . 39
6.5 Limitations of the Evaluation . . . 40

7 Conclusion **41**
7.1 Future Work . . . 41

Bibliography **43**

Chapter 1

Introduction

This thesis presents work done on the application of natural language processing techniques to Swedish court documents and the main work of this thesis was done during the spring of 2016.

In recent years, significant progress has been made to digitalization of Swedish court documents. Since the Swedish court system contains many documents this leaves a vast amount of information to be made available and sorted through.

There are many ways of making information easily accessible by using natural language processing techniques. You could make them searchable through indexing the documents as well as summarizing important key components and topics. To be able to search through the court documents and to summarize them there is a need for identifying and extracting keywords that represents the court documents. This is the domain that this thesis will deal with.

1.1 Problem Statement

A Swedish court document consists of several sections such as plaints, main hearings, rulings and appeals. These are all written with a specific document structure and use legal terminology. From these documents the thesis aims to find a way to compactly describe, for each legal case, the final ruling and the grounds leading up to that ruling with the use of keywords. To describe this compactly descriptive keywords will be used and a compact description is defined as limiting the number of keywords that represents a document to ten. The limitation of ten keywords were empirically selected.

The question, as described above, the thesis aims to solve can be reduced to answering the question *"Can you compactly describe the key components of a Swedish court document automatically using natural language processing (NLP) techniques with a high enough quality to use without human supervision?"*.

1.2 Purpose

This thesis was commissioned by Findwise AB and its purpose is to lay grounds for, in the future, being able to create a predictor for rulings within the court system. The predictor should base its predictions for rulings on its preexisting grounds and previous rulings where similar key components have been identified.

1.3 Objective

The objective of this thesis is to implement one or more algorithms that, with a high quality and low rejection rate, can extract and identify keywords that compactly describes and represents the key components of a Swedish court document. This will be done by dividing the problem into two smaller problems, first identifying possible keyword extraction algorithms to be implemented and tested and second identifying certain preprocessing methods needed to be able to use the keyword extraction algorithms on documents written in natural language.

1.4 Delimitations

This thesis will only focus on the implementation and presentation of the extraction of keywords within NLP for Swedish court documents. If necessary it will use preexisting technology and tools for identifying certain linguistic features.

1.5 Ethical, Social and Sustainability Concerns

Dealing with Swedish court documents and identifying key components in them automatically using NLP have a few ethical and social concerns. The firstmost is that of using Swedish court documents as data. Court documents are public documents however certain court documents are protected by a Swedish law called Sekretesslagen that is a law about confidentiality. This means that before the records can be released to a person there has to be a confidentiality review of each court document. For this reason the data, i.e. the court documents, used should be kept confidential and not be published in the context of this thesis.

Secondly, the court documents used are from different parts of the Swedish legal system and the documents could be unconsciously written or judged in subjective ways therefore the key components identified, of what essentially are the same cases, could be different. This could have an impact on society if used in the future as a predictor.

The court documents used in this thesis are not anonymised. This leaves the possibility of for example a name or company relating to the court document to be used as a possible keyword, even if countermeasures are taken against it. This would

be inappropriate since the keywords used to describe a court document should be anonymous and not tie back to a specific person or company. To keep the integrity of the parties pertaining to the court documents the set used in the evaluation were manually anonymised.

The future use of having a predictor for court rulings could have an economic sustainable impact since it could save time and minimize manual labor when predicting rulings or comparing rulings of different trails.

Chapter 2

Theory

In this chapter the relevant theory regarding extracting keywords automatically is explained. Firstly, the general area of Natural Language Processing (NLP) is presented and the areas where techniques for extracting keywords are used. After this different techniques for extracting keywords will be presented followed by different methods to identify linguistic features that enables the ability to extract keywords from text written in natural language.

2.1 Natural Language Processing

Natural Language Processing (NLP) is a field of science concerning the interaction between computers and natural language. It can be used in varying areas to help automate different tasks such as automatic summarization, sentiment analysis and information retrieval as well as subtasks within these, and other, areas.

2.1.1 Information Retrieval

Information Retrieval (IR) is a large area within NLP and encompasses many of the other subareas of NLP. IR is the activity of obtaining relevant information from a collection of documents. This is usually done through indexing a collection of documents to make them searchable electronically. Indexing is the process of extracting representative terms of a document and the terms extracted from the documents are also known as keywords [1].

An IR system enable the user to sift through relevant information among a vast quantity of information at a faster rate than performing the task manually. There are many different methods used in IR and they can be divided into three main areas; probabilistic, algebraic and set-theoretic [2]. One algebraic method is the Vector Space model that uses one of the most common methods in indexing, i.e. extracting keywords, which is Term Frequency - Inverse Document Frequency (TF-IDF) and this method is explained in Section 2.2.1[3].

2.1.2 Automatic Summarization

Methods to automatically summarize text written in natural language try to solve the problem of finding relevant information in a short amount of time when sorting through a large amount of information or documents. The relevant information can either be summarized concisely using representative keywords or more lengthily by sentences or paragraphs [4].

When automatically summarizing information three steps are usually involved; understanding the content of the document, identifying the most important pieces of information contained in it and writing up this information [5]. The methods used in automatic summarizations can be divided into two types, the first is extraction-based and the second is abstraction-based.

Extraction-Based Summarization

Extraction-based summarization methods are based on selecting important phrases, sentences, paragraphs etc. from the original document and presenting it in a short and concatenated form [6]. These methods are based on statistical and linguistic features of the document and use two steps. The first step is the preprocessing step which is the task of identifying linguistic features and methods and can utilize the techniques mentioned in Section 2.3. The second step is to identify and extract the important topics of the document by extracting keywords which can utilize the techniques mentioned in Section 2.2.

Abstraction-Based Summarization

Abstraction-based summarization methods are based on understanding the original document and recapping it with fewer words [6]. These use linguistic methods to examine and interpret the document and try to find new concepts and expressions that best convey the most important information in the original document. This is done by generating a new text, which describes the concepts and expressions of the important parts.

One tool utilizing both extraction- and abstraction-based summarization methods is SUMMARIST [7] which summarizes text through three steps. First the input text is preprocessed using a few of the linguistic features mentioned in Section 2.3 and then it goes through the first step of identifying the topic, which is an extraction-based summarization method. When the topic has been identified and extracted the tool needs to interpret the topic. Interpreting the extracted topics will enable the system to figure out what the important topics might mean and use this in the last step of generating an abstract summary based on the topics identified.

2.2 Keyword Extraction

Keywords or keyphrases for documents can be used to summarize the documents concisely or as an index used in search engines. This can be done either through assignment or extraction. In keyword assignment the assumption is that all potential keywords appear in a set of predefined keywords, i.e. categories. Keyword extraction is not restricted to a set of keywords from a predefined vocabulary. Instead any keyword identified can be extracted and used as a keyword for a document. [4]

In the following Sections 2.2.1 - 2.2.3 three different approaches to identifying keywords will be presented.

2.2.1 Statistical-Based Approaches

Statistical-based approaches can be used as a weighting scheme for other keyword extraction algorithms but also as a keyword extraction algorithm on its own. In this section, two different techniques for extracting keywords are presented.

Term Frequency - Inverse Document Frequency

Term Frequency - Inverse Document Frequency (TF-IDF) is a statistical measure used to reflect how important a word is to a document in a collection or corpus. It is often used as a weighing scheme in information retrieval and text mining [3].

The TF-IDF measure is a combination of two statistics, the first is the Term Frequency (TF) and the second is the Inverse Document Frequency (IDF). The measure is based on how often the term t appears in document n as well as in the whole collection of documents N. It is calculated as seen in Equation 2.1.

$$tfidf(t, n, N) = tf(t, n) \times idf(n, N) \tag{2.1}$$

The term frequency is calculated as a score between a term t and a document $n \in N$. There are variations in which this is calculated and some of the regularly used functions can be seen in Equation 2.2 - 2.3 [8].

$$tf(t, n) = \begin{cases} 1 & \text{if } t \in n \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

$$tf(t, n) = \sum_{w \in n} \begin{cases} 1 & \text{if } w = t \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

The inverse document frequency is the measurement of whether the term is common or rare across all the documents in the collection or corpus. It is a logarithmic value

of the inverse fraction of n, the documents where the term t occurs, and the whole collection N. Two examples of regularly used functions can be seen in Equation 2.4 - 2.5 [8].

$$idf(n, N) = \log(\frac{|N|}{|n|}) \tag{2.4}$$

$$idf(n, N) = \log(\frac{|N| - |n|}{|n|}) \tag{2.5}$$

The measure for TF-IDF can be used for extracting keywords but needs a predefined corpus to be useful. It will find words that are common for one document but are not common in the whole corpus. If this measure is used on a single document only the term frequency will be taken into consideration. This method is used as a first step in the SALOMON method [9] to find similar rulings of the alleged offenses and opinions of the court from Belgian legal cases. The alleged offenses and opinions of the court are represented as a vector weighted by the TF-IDF measure for each paragraph. The paragraphs are then grouped on mutual similarity by computing the cosine of the angle between the vector representations of the paragraphs. The grouped paragraphs can then be used to identify topics that can be used as keywords by using clustering methods.

The TF-IDF measure is also utilized by Agnolonu, Bacci and Sagri [10] to extract legal keywords from Italian civil case law. The cases were preprocessed using a lemmatizer and POS tagger and then TF-IDF was applied to identify important terms for each civil case in the collection. These important terms are generic and non-specific to the legal domain. To identify keywords for the legal domain a vocabulary list with legal terminology was used to weigh the TF-IDF method into learning that keywords within the legal domain are more important than those that are generic.

N-grams

N-grams can be used to find candidate keywords or keyphrases to be used for a document. N-grams are a continuous sequence of n items, which can be for example words or letters, from a given sequence of text or speech. An n-gram of size one is called *unigram*, of size two *bigram*, of size three *trigram* etc. [11]

The n-gram model predicts the probability of a given sequence of items and is based on conditional probability $P(w_i|w_{i-(n-1)}^{i-1})$ of item w_i following the $w_{i-(n-1)}^{i-1}$ items. The probability can be estimated using the maximum likelihood estimation (MLE) and is calculated by getting the counts $C(w_{i-(n-1)}^{i-1})$ of the n-gram appearing in the

corpus and then normalizing it between 0 and 1. This can be seen in Equation 2.6.

$$P(w_i|w_{i-(n-1)}^{i-1}) = \frac{C(w_{i-(n-1)}^{i-1}w_i)}{C(w_{i-(n-1)}^{i-1})} \tag{2.6}$$

2.2.2 Machine Learning-Based Approaches

Machine learning-based approaches can be used for extracting keywords of relevance. In this section, two different approaches will be presented.

Unsupervised Learning

Cluster analysis is the unsupervised learning technique of grouping together a set of objects in a way such that the objects in one group, called a cluster, is more similar to each other then an object in another cluster [12]. It can be used together with a statistical measure as TF-IDF as used in the SALOMON method [9]. By analyzing a cluster representation of a document similar sentences and words can be found and grouped together. The cluster can then be used to identify a representative topic for the cluster and the topic can be used as a keyword to represent the document.

One common cluster analysis algorithm is the centroid-based algorithm that is also called k-means clustering. K-means creates a cluster based on a centroid, which is the mean of the group of objects, and can be applied to a continuous n-dimensional space. K-medoid is similar but based on a medoid, which is the most representative object for a cluster, and can be used for a wide range of data since it only requires a proximity measure between points. The centroid for the k-means algorithm almost never corresponds to an actual point while a medoid, by definition, is an actual point out of the data. [12]

The most common realization of the k-medoid clustering is the Partitioning Around Medoids (PAM) algorithm, which is a greedy algorithm, and it works as follows:

```
Initialize: Randomly select k of n data points as the medoids and associate
each data point to the closest medoid.
while the cost of the configuration decreases do
    foreach medoid m do
        foreach non-medoid data point o do
            Swap m and o, recompute the cost (sum of distances of points to
            their medoid)
            if the total cost of the configuration increased then
                undo the swap
            end
        end
    end
end
```

The PAM algorithm is used in the earlier mentioned SALOMON method [9] and to find the most representative vector for the paragraphs in the document. The final medoids from the algorithm are the paragraphs that are representative of the document. They are therefore said to represent the topics treated in the alleged offences/opinions of the court and they could be used as keywords to represent the document.

Supervised Learning

A Support Vector Machine (SVM) is a supervised learning technique and can be used to extract and classify whether or not a keyword is good or bad. The keywords can be extracted using a statistical method such as n-grams and the candidate keywords extracted using n-grams can then be classified in a SVM to choose which candidate keywords to use. This is a method that has been employed by Zhang, Tang and Li [13].

SVM is a supervised learning method and given a set of labeled training data belonging to one out of two categories, in this case the two categories good or bad keyword, builds a model based on it and then uses this model to assigns new data into one out of the two categories. We let the training data set be $\{(x_1, y_1), ..., (x_n, y_n)\}$ where x_i denotes an example, a feature vector which in this case is the vector representing the candidate keyword, and $y_i \in \{-1, 1\}$ denotes the classification label. We use this information to attempt to find an optimal separating hyperplane that maximally separates the two classes of training data set. The instances of training data that lay closest to the hyperplane are called *support vectors*. The separating hyperplane corresponds to a classifier and linear SVM and an example of this can be seen in Figure 2.1.

If the training data set cannot be linearly separable then the linear SVM can be extended to a nonlinear SVM using kernel functions such as Gaussian and Polynomial

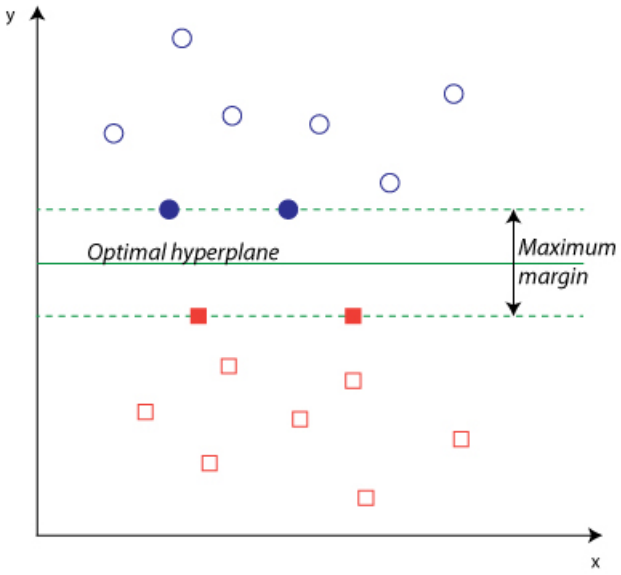

Figure 2.1. An example of a linear SVM.

kernels.

2.2.3 Graph-Based Approaches

KeyGraph is a graph-based keyword extraction technique created to express the main points of the author, and not which words the author uses frequently. This is done by using the information contained in the document and does not need the use of a corpus. The idea of KeyGraph is that each document is written to carry out a few points and these points are related to the terms used in the document. [1] [14]

KeyGraph is based on three major phases; building the graph, adding nodes to the graph which connects the subgraphs and lastly extracting the nodes which are a key component that strongly connects the subgraphs.

The graph G is constructed by choosing the k top terms, i.e. candidate keywords, which occurs with a high frequency within the document D as nodes. The nodes are then connected by how often each pair of nodes $w_i, w_j \in G$ co-occur together within each sentence $s \in D$. An example of a graph constructed in this way can be seen in Figure 2.2.

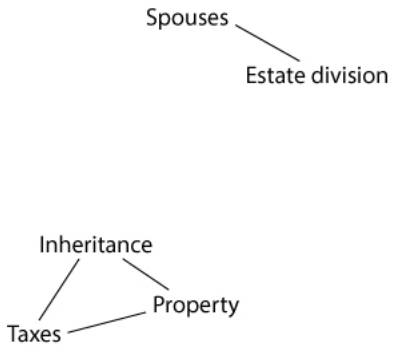

Figure 2.2. An example of a constructed graph in the first step of building a KeyGraph.

The constructed graph G often contains several subgraphs and the maximal connected subgraphs $g \subset G$ are also called clusters. For each cluster we now want to find terms that tie the clusters, i.e. subgraphs, together. The terms that tie these clusters together are called key terms and only the k top ranked key terms in the document D are added to the graph G. These terms are then connected to the subgraphs if they connect two or more clusters by calculating how often they, together with each high frequency term in G, co-occur within each sentence $s \in D$. An example of adding key terms to the constructed KeyGraph in this way can be

seen in Figure 2.3.

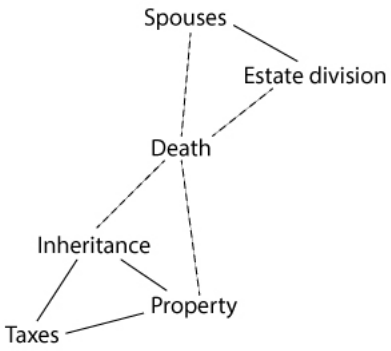

Figure 2.3. An example of connecting constructed subgraphs with key components.

From the final graph the n nodes that connect to two or more clusters and have the highest total score of all its connections are chosen to represent the documents as keywords.

2.3 Preprocessing Methods

Semantic, syntactic and orthographic features and methods within NLP are commonly used in keyword extraction systems. In this section different techniques for identifying preprocessing methods, which can be used before the application of keyword extraction techniques, are presented.

2.3.1 Part-of-Speech Tagging

A Part-of-Speech tagging (POS tagging) is a method that assigns part of speech to each word, such as noun, verb and adjective, in a text or a document [15]. A word on its own may have several meanings, for example the word store may be either a noun, a finite verb or an infinitive. However, when taking into account the context it is used the word will only have one meaning. By using a POS tagger you can identify these and it is a method used in noun phrase chunking to identify the noun phrases [16]. An example of how a POS tagger would work is if the sentence "The Supreme Court establishes the Court of Appeals decision" was POS tagged, which would yield the results shown in Figure 2.4.

Figure 2.4. An example of POS tagging.

2.3.2 Noun Phrase Chunking

Text chunking is the method of dividing a text or a sentence into small segments, which are syntactically related words [16]. These become a member of the same phrase and are non-overlapping; hence one word can only be a part of one phrase. The method of noun phrase chunking (NP chunking) is therefore the method of dividing the text or sentence into phrases that are nouns. An example of NP chunking used on the sentence "The Supreme Court establishes the Court of Appeals decision" is shown in Figure 2.5.

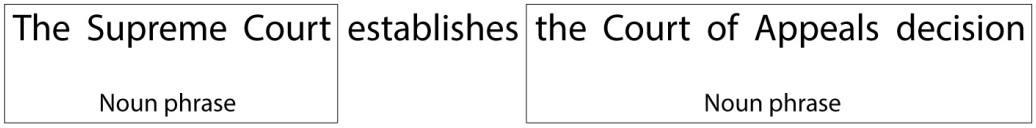

Figure 2.5. An example noun phrase chunking.

2.3.3 Stemming

Documents, written in natural language, that are grammatically correct usually contains words written in different forms. For example the word "organize", which has the present form "organizes", present participle form "organizing" and past participle form "organized" however they all stem from the same word. There are also families of related words such as democracy, democratic, and democratization, which do not stem from the same basic word form but are still related and should therefore be stemmed to the same base. To find the words of a document that stem from the same form or words that are related to each other a technique called stemming can be used. Stemming is a crude method that chops off the end of the word to reduce inflectional forms and sometimes related forms of a word to a common base form. This technique can be useful when finding how common or uncommon a word is to a document. [3]

2.3.4 Lemmatization

Lemmatization is a method similar to stemming but it utilizes the vocabulary and morphological knowledge of each word. For each word in a document it finds the words base or dictonary form by removing the inflectional ending, also known as a

lemma. For example, if the word “saw” appeared in a document a lemmatization of the word would yield the base form “see”. [3]

2.3.5 Co-occurrence

Co-occurrence refers to a pair of words that co-occur in a large number of documents. It could also refer to a pair of words that although they co-occur in a small number of documents they occur close to each other within those documents. Co-occurrence can therefore be used as an indicator of semantic proximity or an idiomatic expression. [17] [18]

2.3.6 Stop Words

Stop words are used within NLP to filter out words that are common in all texts before processing the text [19]. Using this method would guarantee that words without any meaning, such as “the”, are not used as candidate keywords.

The TF-IDF method could also be used to remove common words that occur frequently in the document as well as in the corpus and therefore give it a low score and eliminate the probability of it being a candidate keyword. However the collection used does not guarantee that common words occur frequently in the corpus and could therefore be given a high score and seen as a candidate keyword. Using a stop word list instead is an easier and more certain way of eliminating specific common words as candidate keywords.

2.4 Conclusions of the Literature Study

From the related work and the different approaches to extracting keywords studied it was decided that in order to cover all possible approaches a method from each of the three categories were to be implemented. Out of the statistical-based approaches the TF-IDF method was chosen to be implemented since it takes into account how important a word is to a document and its collection which could be of use since this thesis uses a very specific domain and vocabulary. This could mean that certain legal vocabulary could be identified as unique or common for the document and the whole collection. The n-gram method, however, does only take into account the probability of a word appearing in the whole collection. This means it does not distinguish between how a word is used in the document it is supposed to represent as well as how the word is used in the collection.

Using the TF-IDF method with a legal vocabulary which could weigh legal specfic words as more important was excluded from this report since it was deemed to take too much time to create because a vocabulary in a usable format did not yet exist.

In the second category, there are two methods mentioned, one that is an unsupervised learning method and another which is a supervised learning method. The

supervised learning method has a need for keywords preannotated as “good” or “bad” keywords. There are no data containing this already and therefore deemed to take up too much time for this thesis to be a feasible approach. The unsupervised learning method was therefore chosen instead of the supervised learning method to be implemented for this thesis.

In the third category of graph-based approaches the KeyGraph method was chosen to be implemented since it is the only one of the methods which is not dependent on a whole collection of court documents and only takes into account important word in the document itself.

Two preprocessing methods were also chosen to be used and the first one chosen is that of using stop words. This is an easy and fast way of filtering out words that lack any meaning and are only used in conjunction with other words. The second feature chosen was that of stemming since all three keyword extraction methods rely on the frequency of a word appearing in a document there was a need to identify the base of a word to make sure that words written in different forms are counted as a part of its frequency. In this case a stemmer, which does this approximately according to a few rules, were chosen over a lemmatizer since it needs a dictionary and is therefore a very time consuming task.

The other two preprocessing methods, POS taggning and NP chunking, were not chosen to be implemented in this thesis since they were deemed to take too much time to implement and use properly.

Chapter 3

Method

In this chapter the overall architecture of the keyword extraction system and the experiments performed with it are described. Firstly the architecture of the system will be presented followed by a description of the data set used. Then the implementation of the lexical analysis, preprocessing methods and keyword extraction techniques used will be described. Lastly there will be a description of how the top keywords extracted for the systems were evaluated.

3.1 Architecture

The keyword extraction system was created using the Findwise i3[1] architecture. i3 is a tool that can be used to index documents and allows its user to create a pipeline which enables the possibility of processing each document individually. The result of the processing can then be used to for example write to a file or a search engine, for example Solr[2] or Elastic[3].

The i3 architecture also allows the user to control the pipeline using a GUI. Since each step in the i3 pipeline works as individual components that are independent of each other this allows the user to shut on and off any component at any time. It also allows the user to alter parameters and run tests automatically and on a timer with ease.

The pipeline used in the architecture for this thesis was divided into seven main components and these can be seen in Figure 3.1.

[1] http://findwise.com/technology/findwise-i3
[2] http://lucene.apache.org/solr/
[3] https://www.elastic.co/

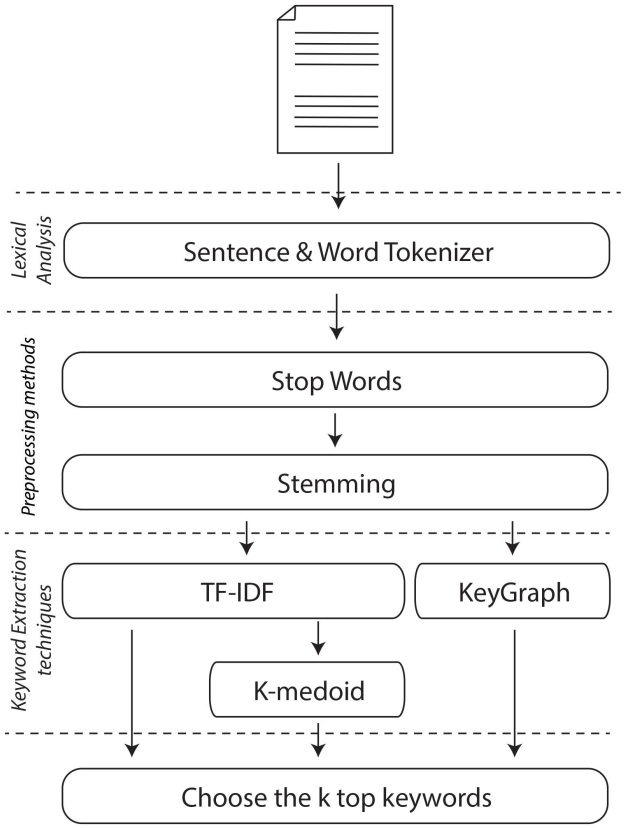

Figure 3.1. An overview of the i3 architecture used in this thesis to extract keywords.

3.2 Data Set

The set of data used are about 4500 Swedish court documents that have been scanned using OCR. Since the documents were scanned using OCR they contain some structural errors such as a "k" turning into an "lc" and some words having spaces between each letter. These structural errors were manually corrected for the documents used in the test.

Each court document contains all the relevant information for its case; appellant, appellee, matter, appeals, rulings, accounts, demands, grounds for rulings and how to appeal. The case also contains this for all instances of court the case has been raised in. The cases used are within the following 11 categories:

- Inheritance and gift tax
- Property tax
- Wealth tax

- Income tax
- International tax law
- Value added tax
- Excise tax
- Tax payment
- Tax surcharge
- Social-security contribution
- Tax dispute

The data set used have been divided into two sets, one used for training the data and tweaking parameters and the other is used for testing and evaluation.

3.3 Lexical Analysis

Before preprocessing a document the document needs to be analyzed lexically. This was done through two different types of tokenizations and those are described here.

3.3.1 Word Tokenization

Word tokenization is the task of chopping up a document into pieces, tokens, where each token represents a word in the document. The word tokenizer used is called ClassicTokenizer[4] and works well with most European languages. It identifies words as tokens by the following four rules:

- Tokens ending with a whitespace.
- Tokens ending with a punctuation, where the punctuation is removed. If the punctuation is not followed by whitespace then the punctuation is considered a part of the token and therefore not removed.
- Tokens ending with a hyphen, unless the token contains a number, in which case the whole token is interpreted as a product number.
- Recognize email addresses and internet hostnames as one token.

Using ClassicTokenizer allows for dates and law references in the court documents to be seen as a single word by the tokenizer.

[4] `https://lucene.apache.org/core/4_0_0/analyzers-common/org/apache/lucene/analysis/standard/ClassicTokenizer.html`

3.3.2 Sentence Tokenization

Sentence tokenization is the task of chopping up a document into tokens where each token represents a sentence within the document. The sentence tokenizer used is called BreakIterator[5] and it was used to analyze the sentence boundaries in the document. It allows for selecting sentences with the correct interpretation of periods within numbers and abbreviations as well as trailing punctuation marks such as quotation marks and parentheses.

3.4 Preprocessing

Before being able to use the keyword techniques on the tokenized words and sentences two preprocessing methods needed to be used on the words and sentences. These two preprocessing methods are described in this section.

3.4.1 Stop Words

The preprocessing method of filtering out common words from the document using a stop word list was the second step in the i3 pipeline implemented. This allowed words that contain no relevant meaning to the context to be filtered out, and this enabled common function words such as "i" (in), "på" (on) and "om" (about) to be removed from the possibility of being candidate keywords. The list of Swedish stop words[6] used was that recommended by the stemming algorithm Snowball[7]. The list of stop words used can be seen in Figure 3.1.

During the testing of the stop word list together with the keyword extraction techniques names of people were shown the be chosen as keywords. Therefore the most common Swedish names according to the SCB[8], Statistics Sweden, were added to the stop word list to filter these out as possible keywords.

[5] `https://docs.oracle.com/javase/7/docs/api/java/text/BreakIterator.html`
[6] `http://snowballstem.org/algorithms/swedish/stop.txt`
[7] `http://snowballstem.org/`
[8] `http://www.scb.se/namnstatistik/`

och	det	att	i	en	jag	hon	som
han	på	den	med	var	sig	för	så
till	är	men	ett	om	hade	de	av
icke	mig	du	henne	då	sin	nu	har
inte	hans	honom	skulle	hennes	där	min	man
ej	vid	kunde	något	från	ut	när	efter
upp	vi	dem	vara	vad	över	än	dig
kan	sina	här	ha	mot	alla	under	någon
eller	allt	mycket	sedan	ju	denna	själv	detta
åt	utan	varit	hur	ingen	mitt	ni	bli
blev	oss	din	dessa	några	deras	blir	mina
samma	vilken	er	sådan	vår	blivit	dess	inom
mellan	sådant	varför	varje	vilka	ditt	vem	vilket
sitta	sådana	vart	dina	vars	vårt	våra	ert
era	vilkas						

Table 3.1. The list of words used as stop words.

3.4.2 Stemming

The second preprocessing method used was that of finding the stem of each word. This was to make sure that the frequency of how often different words appeared included the different forms of a word. The stemming algorithm used as the second step to get the stem of each word is called Snowball [20]. This algorithm is adapted to work well on different European languages. The version used is adapted to the Swedish language and takes advantage of the semantic rules in the language. This is used to form rules which give the stem of each word.

The algorithm first defines that the Swedish alphabet includes the letters a - ö. It also defines that the letters a, e, i, o, u, y, ä, å, ö are vowels. A region R1 is used which is the region after the first non-vowel, i.e. consonant, number, punctuation and other non-vowel characters, following a vowel, or is the null region at the end of the word if there is no such non-vowel. The region R1 also has the condition that it needs to contain three letters before it starts. An example of this would mean that the word *bolag* (company) gives us the R1 region *ag*. The algorithm also defines a valid ending of a word, also called **s**-ending, as one of the letters b, c, d, f, g, h, j, k, l, m, n, o, p, r, t, v, y.

The algorithm then needs to go through the following three steps.

1. Search for the longest among the following suffixes in R1, and perform the action indicated.

    a) **a, arna, erna, heterna, orna, ad, e, ade, ande, arne, are, aste, en, anden, aren, heten, ern, ar, er, heter, or, as, arnas, ernas,**

ornas, es, ades, andes, ens, arens, hetens, erns, at, andet, het, ast

delete

b) **s**

delete if preceded by a valid **s**-ending (the letter of the valid **s**-ending does not necessarily have to be in R1)

2. Search for one of the following suffixes in R1, and if found delete the last letter.

   dd, gd, nn, dt, gt, kt, tt

   For example, friskt → frisk.

3. Search for the longest of the following suffixes in R1, and perform the action indicated. For example the word *trolig* longest suffix would be *lig* and not *ig* and therefore yield the stem *tro* and not *trol*.

   a) **lig, ig, els**

   delete

   b) **löst**

   replace with **lös**

   c) **fullt**

   replace with **full**

After going through each of these three step the stem of the word is gotten. An example of this can be seen in Table 3.2.

word	stem	word	stem
bolag	bolag	fastställ	fastställ
bolags	bolag	fastställa	fastställ
bolagen	bolag	fastställas	fastställ
bolagens	bolag	fastställer	fastställ
arv	arv	fastställs	fastställ
arven	arv	fastställande	fastställ
arvens	arv	fastställandes	fastställ

Table 3.2. An example of words and their stemmed form using the Snowball algorithm.

3.5 Keyword Extraction Techniques

Three different techniques for extracting keywords were implemented and added as three separate steps in the i3 pipeline. In this section a description of how they were implemented is presented.

3.5.1 Statistical-Based Approach: TF-IDF

TF-IDF was the first technique implemented and used to statistically extract keywords from the court documents. Firstly the term frequency calculation was implemented and a logarithmic scale was used. The logarithmic scale, as seen in Equation 3.1, was used to normalize the term frequency of the documents and the corpus.

$$tf(t,d) = 1 + \log(f_{t,d}) \tag{3.1}$$

$f_{t,d}$ is used to denote the raw frequency, i.e the number of times a term t occurs in document d. Then the inverse document frequency is calculated using Equation 3.2 where t is the term and D is the set of all documents.

$$idf(t,D) = \log \frac{|D|}{|d \in D : t \in d|} \tag{3.2}$$

Both the functions presented are then multiplied as seen in Equation 3.3 to give a score for each word which represents how important a term is to the document while taking into account the set of all documents.

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D) \tag{3.3}$$

The words with the highest score from the TF-IDF algorithm were those deemed, by the keyword extraction technique, as those most likely to be keywords. Out of these the top 10 words were chosen to represent the court ruling and grounds for the ruling as keywords.

3.5.2 Machine Learning-Based Approach: K-medoid

The second approach implemented was that of the unsupervised learning method K-medoid. The K-medoid technique is based on representing each sentence in the court documents as a vector. The vector contains information of the weighted score which is calculated using the TF-IDF algorithm as described earlier in Chapter 3.5.1.

The K-medoid method implemented use the Partitioning Around Medoids (PAM) algorithm and it goes through the following three steps using all of the n data points, i.e the vectors mentioned earlier.

1. Initilazation: Randomly select k of the n data points as medoids.
2. Associate each data point to the closest medoid by calculating the similarity between the point and all the medoids.
3. For each medoid m and each data point o associated to m, swap m and o and compute the total cost of the configuration (the average dissimilarity of o to all the data points associated with m). Then select the medoid o with the lowest cost of the configuration.

Step 2 and 3 were repeated until there were no more changes in the assignments. To calculate the similarity and distance between each medoid the cosine measurement was used. It was calculated using Equation 3.4 where $\vec{A}$ and $\vec{B}$ are the vector representations of the weighted terms in each sentence.

$$similarity = \cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{||\vec{A}|| \cdot ||\vec{B}||} \tag{3.4}$$

To get the top 10 representative sentences from the K-medoid k were set to 10. For each sentence the word with the highest weighted score were chosen to represent that sentence which gave us the top 10 extracted keywords chosen to represent the court ruling and grounds for the ruling.

3.5.3 Graph-Based Approach: KeyGraph

The last approach to extract keywords implemented is that of the KeyGraph algorithm. The implementation of the KeyGraph algorithm is divided into four steps. The first step of the algorithm is to populate the KeyGraph G with nodes. The nodes are created from the k most frequently used words w in the document D. The words frequency score is calculated using Equation 3.1 and the k most frequently used words had the highest scores. These k words are also called High Frequency Term (HFT) and are seen to represent concepts of the document and therefore make up the nodes of the KeyGraph G as demonstrated in Figure 3.2.

The second step in the algorithm is to connect the nodes, i.e. HFT, in the KeyGraph G by how strongly associated they are with each other. The association between each pair of nodes $w_i, w_j \in G$ means how often the words occur together in the same sentence s. The association is calculated using Equation 3.5. If an association value is larger than 0 that means that the two nodes $w_i, w_j \in G$ are connected. All

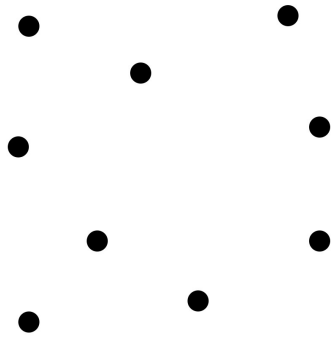

Figure 3.2. Demonstrates the KeyGraph algorithm first step of adding the k top frequent words of the document as nodes in the graph G.

the nodes that connect to other nodes will create connected subgraphs g as can be seen in Figure 3.3.

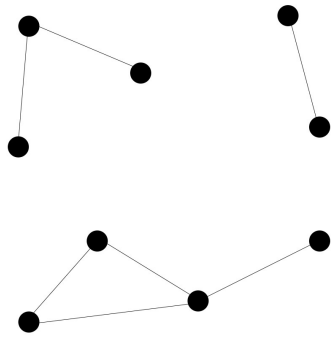

Figure 3.3. Demonstrates the KeyGraph algorithm second step of connecting the nodes if they have an association by co-occuring in the same sentences to create connected subgraphs $g \subset G$

$$assoc(w_i, w_j) = \sum_{s \in D} min(|w_i|_s, |w_j|_s) \tag{3.5}$$

The third step in the algorithm is to find connections between each subgraph in G. This is done by finding terms in the document D which can work as bridges, i.e. occurring in the same sentences s as the nodes in the subgraphs, between each subgraph $g \subset G$. These possible terms are called a High Key Term (HKT), and

the probability of a term $w \in D$ being a HKT and connecting a subgraph $g \subset G$ is calculated using Equation 3.6.

$$key(w) = 1 - \prod_{g \subset G} (1 - \frac{based(w, g)}{neighbors(g)}) \tag{3.6}$$

$$based(w, g) = \sum_{s \in D} |w|_s |g - w|_s \tag{3.7}$$

$$neighbors(g) = \sum_{s \in D} \sum_{w \in s} |w|_s |g - w|_s \tag{3.8}$$

$$|g - w|_s = \begin{cases} |g|_s - |w|_s & \text{if } w \in g \\ |g|_s & \text{if } w \notin g \end{cases} \tag{3.9}$$

The fourth and final step is to extract 10 candidate keywords with the highest probability of representing the court ruling and grounds for the ruling. Firstly the top k HKT are added to the graph G and then the association between the HKT w_i and HFT w_j are calculated by Equation 3.5. The connection of HKT and HFT can be seen in Figure 3.4. The nodes in the KeyGraph G which connect two or more clusters and have the highest sum of the $assoc(w_i, w_j)$-score are the nodes which represents the document the best according to the KeyGraph. They are therefore the ten keywords which are chosen as the top keywords given by the algorithm.

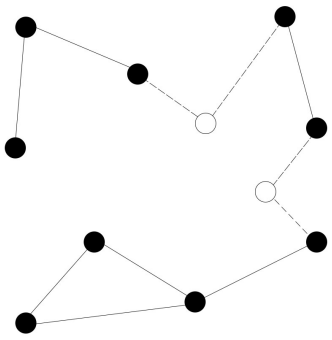

Figure 3.4. Demonstrates the KeyGraph algorithm third step of connecting the connecting the subgraphs $g \subset G$ by finding words which co-occur with the nodes in the subgraphs together in sentences s.

During testing of the KeyGraph algorithm k was empirically chosen to 30.

Chapter 4

Evaluation

In this chapter the evaluation of the three keyword extraction techniques will be described. This was done through a human evaluation which focuses on the quality of the extracted keywords.

4.1 Subjects

The subjects were recruited from the circle of acquaintances for the author of this report. 10 subjects were recruited, out of which five were male and five female. One of the subjects was a law school student and the rest were of varying educational backgrounds and non-experts. The objective of the evaluation was first to get more experts to participate but this proved difficult since none of the contacted experts and institutions with experts wanted to participate in the evaluation. Therefore, non-experts were included in the evaluation.

The first language for all the participating subjects was Swedish. The youngest subject was 22 and the oldest was 57.

4.2 Experimental Documents

A set of four court documents from the test data set were used for the evaluation. The four documents where deemed to take about 30 minutes for a non-expert to evaluate and therefore to make sure more non-experts could participate no more documents were used in the evaluation.

From the four court documents the part pertaining to the ruling and the grounds for the ruling were the only parts of the documents used. The purpose of this was to make sure that the subjects only assessed the part of a court document which the keywords represents.

All Swedish court documents are public documents, except some who are protected

by *Sekretesslagen*, which is a Swedish law about confidentiality. Therefore there has to be a confidentiality review of each document before they are handed out. The four documents used in the evaluation have therefore been anonymised.

4.3 The Design of the Evaluation

The evaluation was designed to allow the subjects to be able to participate online and it was written in Swedish. The subjects were given instructions that the evaluation pertains to evaluating the quality of how well the keywords presented in the evaluation represent the extracted ruling and grounds for the ruling from court documents.

The evaluation was then designed to allow the subjects access to only one court document and its keywords at a time. For each court document the subjects were then instructed to first read the whole document. They were also, for each court document, given a vocabulary describing the legal terms used in the document.

After reading the document the subject was then presented with three groups of keywords and they were then asked to evaluate how well each individual keyword for each group represents the document. The subject where then asked to evaluate how well each group of keywords together represents the document. They were asked to rate this on a scale from one to ten, where one meant the keyword represented the document poorly and ten that the keyword represented the document well. The scale of one to ten were used to give the subjects granularity when evaluating how poor or well a keyword was and to give them the possibility of selecting a middle value which would be neither poor nor well. Each group and its keywords with its rating scale was presented in the form shown in Figure 4.1. The subject indicated its rating by choosing the radio-button with the appropriate value. Subjects could refer back to the document and reread it as often as required.

The keywords were presented in order of how well the algorithms thought they represented the document. The keyword with the heights score were presented top left and the keyword with the lowest score were presented at the bottom right. The subjects were not aware of the ordering of the keywords.

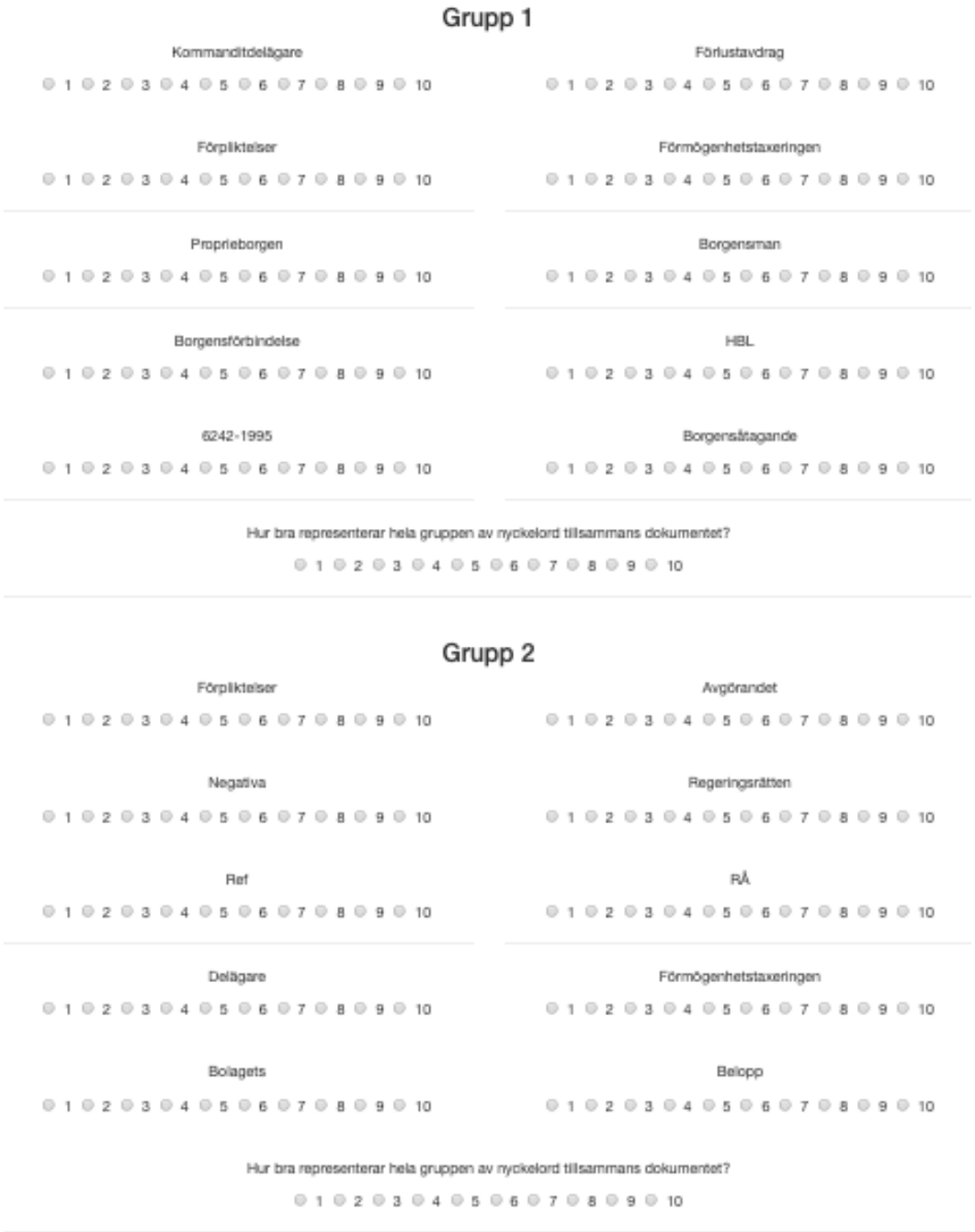

Figure 4.1. An example of the form the keywords and the groups were presented in the evaluation.

Chapter 5

Results

In this chapter the results from the evaluation of the three different implemented keywords extraction techniques, as described in Section 3.5.1, are presented. The results will be presented in three forms, the first is the generated keywords of the algorithms for the four documents used in the evaluation, the second the average and median result and lastly the average deviation of the subjects scoring.

5.1 Generated Keywords

The top ten keywords for each document used in the evaluation differed between the three keyword extraction algorithms. For all four documents neither top ten keywords were exactly the same for each algorithm. Some of the documents had keywords in common for each algorithm but in a different order. However, most of them differed a lot and for one of the documents the K-medoid algorithm couldn't generate ten keywords. The generated keywords for two of the four documents used in the evaluation presented from highest to lowest score of representing the document can be seen in Table 5.1.

The keywords were of varying quality and some of them, for example *kr*, *Ref* or *712*, are not keywords of enough meaning to represent the document as a keyword.

	TF-IDF	K-medoid	KeyGraph
Document 1	Proprieborgen	Kommanditdelägare	Förpliktelser
	3273-2000	Förlustavdrag	Avgörandet
	Borgensåtagande	Förpliktelser	Negativa
	Borgensman	Förmögenhetstax-eringen	Regeringsrätten
	Förlustavdrag	Proprieborgen	Ref
	Ansvarsåtagande	Borgensman	RÅ
	6242-1995	Borgensförbindelse	Delägare
	Förmögenhetstax-eringen	HBL	Förmögenhetstax-eringen
	Kommanditdelägare	6242-1995	Bolagets
	Andelsvärde	Borgensåtagande	Belopp
Document 2	712	Tillämpliga	Belopp
	368	Förfrågan	Bolaget
	603	Frågan	Felaktig
	143	Fullt	Befrias
	404	Hälften	Oskäligt
	Koncernintern	Bolaget	Fullt
	168	712	Avgiften
	Underprisöverlåtelse		Skattetillägg
	Förfrågan		Uppgift
	Fullt		kr

Table 5.1. The top 10 keywords in highest to lowest score for two documents used in the evaluation for each of the three keyword extraction algorithms

5.2 Average and Median Score

From the results of the evaluation the average and median score for each document was calculated by the subjects scoring on a scale from 1 to 10 for each group of keywords as a whole. These groups represented keywords generated by one of the three keyword extraction techniques used. The average and median score can be seen in Table 5.2. From the results the score of $1 - 5.5$ were deemed as a technique which yielded keywords that did not represent the document with enough certainty. If the score was that of $5.5 - 10$ the keywords yielded by that technique were deemed to represent the keywords with enough certainty.

Of the total average and median we can see that the keywords generated by the KeyGraph algorithm was those which on average and median performed the best but yielded results which lie closely around the bottom of the acceptable boundary. The technique which yielded the total worst average and median results were that of the K-medoid algorithm. As can be seen from the average and median score of each document for each technique they all performed better or worse on different

documents. However, from the results we can see that the scoring of TF-IDF and K-medoid closely relate to each other.

From the results we can also see that the median total score is slightly higher than that of the average score. This shows that the majority of the answers by the subjects are the same as or higher than the average score for each keyword technique.

	TF-IDF		K-medoid		KeyGraph	
	Average	Median	Average	Median	Average	Median
Document 1	7	6	7	7	4	3
Document 2	3	3	2	2	5	5
Document 3	5	6	4	5	7	7
Document 4	5	5	6	6	7	8
Total	5	5.5	4.75	5.5	5.75	6

Table 5.2. The average and median score for each group of keywords.

For each individual keyword for each document and keyword technique the rejection rate was on average and median was calculated. The individual keywords with a score below 5.5, as described above, was deemed as non-representative keyword and therefore rejected. The average and median rejection rate for the ten individual keywords for each document and keyword extraction technique can be seen in Table 5.3. As seen from the results the KeyGraph algorithm rejected the least amount of keywords in total but the rejection rate for each document varied greatly from 90% to only 10%. The other two techniques had a slightly higher rejection rate of the individual keywords but the rejection rate on average between the documents did not vary as much as that of the KeyGraph algorithm.

The median rejection rate is slightly higher for the K-medoid algorithm than that of the average rejection rate but in both cases it still rejects the most keywords.

The TF-IDF algorithm median rejection rate is the same as the average rejection rate except for in document 4 where the rejection rate is 10% lower. The same goes for the KeyGraph algorithm where the median rejection rate is 10% lower than that of the average rejection rate for document 1 and 2.

	TF-IDF		K-medoid		KeyGraph	
	Average	Median	Average	Median	Average	Median
Document 1	50%	50%	40%	40%	90%	80%
Document 2	80%	80%	100%	100%	90%	80%
Document 3	70%	70%	90%	90%	10%	30%
Document 4	80%	70%	60%	70%	50%	40%
Total	70%	70%	72.5%	80%	60%	60%

Table 5.3. The average and median rejection rate of each individual keyword.

The ten top keywords generated for each technique were given different scores. The top keywords had the highest score for the given technique and seen by the technique to best represent the document. The bottom keywords had the lowest score and therefore, by the technique that generated them, seem to represent the document less better than the top ones. In Table 5.4 the average score by the subjects from the highest to the lowest scoring keywords per technique and document can be seen. The results per technique and document varies a lot and for some documents the keywords which had the highest score were also given the highest score by the subjects. However, for document 2 and 3 the highest scoring keywords were given lower scores by the subjects. The scores by the subjects do not follow a pattern of the highest scoring keyword by the algorithm also being the highest scoring by the subjects.

	Top / Technique	1	2	3	4	5	6	7	8	9	10
Document 1	K-mediod	8	8	5	7	7	5	5	6	5	6
	KeyGraph	4	3	3	5	2	3	4	7	4	3
	TF-IDF	6	5	6	5	7	4	5	7	8	5
Document 2	K-mediod	3	2	2	2	4	5	2			
	KeyGraph	4	5	5	5	5	3	5	8	3	3
	TF-IDF	2	2	1	2	2	6	2	7	3	2
Document 3	K-mediod	5	4	5	3	6	4	5	3	1	3
	KeyGraph	6	6	6	6	7	7	7	5	7	6
	TF-IDF	5	3	3	4	3	3	7	6	1	6
Document 4	K-mediod	4	3	6	7	1	9	2	4	6	2
	KeyGraph	6	7	6	7	3	5	6	2	5	4
	TF-IDF	6	5	1	5	4	2	1	2	8	2

Table 5.4. The average score for each individual keyword per document and technique. The Top 1 keyword is the keyword which had the highest score for the technique and Top 10 the keyword had the lowest score.

5.3 The Average Deviation

From the evaluation the average deviation between each subject's scoring on each question was calculated. Then the total average deviation for each group of questions were calculated and the result of this can be seen in Table 5.5. The results yield that on average the subjects scoring differed approximately ± 2 points. The scoring by the subjects differed the least with the TF-IDF algorithm and the most with KeyGraph algorithm. The difference is however very small.

	TF-IDF	K-medoid	KeyGraph
Document 1	2.27	2.13	1.97
Document 2	1.18	1.55	2.10
Document 3	1.72	1.98	2.14
Document 4	1.72	1.83	2.16
Total Average Deviation	1.73	1.87	2.09

Table 5.5. The average deviation between each answer.

Chapter 6

Discussion

In this chapter the results of the three keyword techniques will be discussed, analyzed and compared.

6.1 Statistical-Based Approach

As described in Section 3.5.1 the statistical-based approach TF-IDF is constructed by first using the preprocessing methods stop words and stemming and then calculating the TF-IDF score on the stemmed and filtered words.

The quality of the keywords extracted using this approach was shown by the results, of both measures, to be in the top of the interval for keywords determined not to represent the documents with enough certainty. It also had a very high rejection rate for the individual keywords within each group. The keywords rejected were both keywords of higher and lower TF-IDF score, i.e. keywords as seen by the algorithm to represent the document more and less. This means that the approach is not an appropriate approach on its own to use as a keyword extraction technique, at least when it comes to the quality of the extracted keywords.

The results from evaluating the quality of the keywords extracted using a statistical-approach could presumably be because it only takes into account how unique a word is to the document compared to the whole collection of documents. Court documents could have domain specific law words which could be frequently used in the whole collection and hence not statistically seen as a word of importance. Therefore some words that should be seen as an important and unique word are actually given a low score by the TF-IDF approach. One possible improvement would be to use a law vocabulary where words in the documents which exist within the law vocabulary is weighted higher. This would allow domain specific words to be valued higher in the TF-IDF scoring than other words and therefore have a higher statistical probability of being candidate keywords.

6.2 Machine Learning-Based Approach

As described in Section 3.5.2 the machine learning-based approach K-medoid is constructed by first using the preprocessing methods stop words and stemming. Then it calculates the TF-IDF score for each candidate keyword and finds the sentences of the document which represents the document best. The keywords which have the highest TF-IDF score is then chosen to represent the chosen sentences.

The quality of this approach is shown by the results to be almost the same as the TF-IDF approach, which it is supposed to improve. In some cases the K-medoid approach even performs worse than the TF-IDF approach. The K-medoid approach has a high rejection rate of the individual keywords and the quality is in the higher part of the lower interval were keywords are deemed to not be of high enough quality to represent the document. The keywords rejected are both keywords seen to represent the document better or less. This approach is therefore not a feasible approach on its own to use as a keyword extraction technique for court documents.

A possible improvement that could increase the probability of extracting domain specific law terminology would be to improve the TF-IDF algorithm with a law vocabulary as mentioned above in Chapter 6.1. The K-medoid algorithm itself might not be improvable. One option would be to choose representative paragraphs instead of representative keywords. This could improve the similarity measures since most sentences used are probably very different. However, some sentences might be added to a cluster which it is not very similar to and this would yield representative sentences which are not representative of the other sentences in its cluster.

Using paragraphs instead of sentences is probably not feasible in the usage of finding keywords which represent the ruling and grounds for the ruling of a Swedish court document since it usually does not contain paragraphs enough to get ten representative paragraphs. It could be used if the candidate keywords are not taken from each individual cluster but candidate keywords with the highest probability out of all the representative paragraphs.

6.3 Graph-Based Approach

The graph-based approach KeyGraph used is described in Section 3.5.3 is constructed by first using the preprocessing methods stop words and stemming. Then it extracts the keywords using the KeyGraphs algorithm.

The results of the evaluation of the keywords extracted using the KeyGraph algorithm showed to be in the interval of acceptable quality for it to represent the rulings and the grounds for the ruling in Swedish court documents. However it still had a very high rejection rate of more than half of the keywords extracted on average and

median which is not acceptable for representing the documents.

The reason why a high rate of the individual keywords were rejected could be due to the HFT chosen to be used in the subgraphs. The HFT decide which HFK should be used to connect the subgraphs since they are based on how the words in the subgraphs co-occur. If other HFT were chosen then there is a probability that other HFK would be used. This could yield that other final keywords are chosen to represent the document. One possible solution to choosing the correct most frequent and important words could be to weigh words that were used in connection with law terminology.

6.4 Comparison

When comparing all three approaches for extracting keywords to represent the rulings and grounds for the rulings in Swedish court documents the KeyGraph approach was the only one with a quality high enough to qualify as an acceptable approach which yields a group of keywords that represents the document. However neither approach had a rejection rate less than 50% which would be an acceptable rate of how many keywords which individually are not a representative keyword. The individual keywords scoring order for each approach showed no conclusive difference between the highest scoring being more representative or the lowest scoring being less representative.

To lower the rejection rate and heighten the quality score for all three approaches a possible improvement would be to use more preprocessing methods. One improvement would be to use preprocessing methods to filter out more words that should not be possible candidate keywords. The preprocessing methods could also make it possible for a group of words, such as noun phrases, to be used together as a keyphrase instead of as separate keywords which are of less importance.

One preprocessing method which has already been mentioned could be using a law vocabulary which would enable law terminology to be valued higher. Another feature would be that of POS tagging, as described in Section 2.3.1, which could help or even replace the usage of stop words. The POS tagger could identify and filter out words such as determiners or prepositions which are words that should not be used as candidate keywords. Determiners and prepositions are usually words which are used as stop words and using a POS tagger to identify these instead would render the usage of stop words unnecessary. The usage of a POS tagger could possibly filter out more words from the list of possible keywords than a stop word list could and therefore could improve the list of candidate keywords to run the actual keyword extraction technique on.

NP chunking, which is described in Section 2.3.2, could be used as another possible preprocessing improvement. This would identify noun phrases that could be used

to make words which together are a better representative keyword then they are apart.

Another possible linguistic feature would be to use a lemmatizer instead of the stemmer. This would yield more accurate results when using the base form of each word instead of an approximate base form since the frequency of a word occurring could differ using stemming because they could use different base forms.

6.5 Limitations of the Evaluation

The evaluation was performed on a small data set and few participating subjects which could have yielded inconclusive or other results than if the evaluation would have been performed on a larger data set, more participants or expert participants. An improvement to get better and more confident results would be to perform the evaluation on a larger data set and more subjects. A complement to the current evaluation could be to have one or more experts evaluate a larger data set then the current evaluation.

Chapter 7

Conclusion

Out of the three approaches used the KeyGraph algorithm yielded results which showed it performed better than the other two. However, none of the approaches generated ten keywords with a high enough quality score and a low enough rejection rate to be used as a possible way to compactly describe the rulings and grounds for the rulings of a Swedish court document.

There are a few possible reasons as to why neither of the three approaches affirmatively answers the question *"Can you compactly describe the key components of a Swedish court document automatically using natural language processing (NLP) techniques with a high enough quality to use without human supervision?"*. These reasons are discussed and analyzed in Chapter 6 and four possible improvements were mentioned which could improve the quality score and lower the rejection rate for the three approaches. These four preprocessing methods are law vocabulary, POS tagging, lemmatizer and NP Chunking.

To evaluate the quality of the keywords with a higher certainty the number of participants should be higher than in this thesis. More participants who have a higher knowledge of law terminology could also improve the accuracy of the evaluation.

In conclusion the results show that all three approaches need to be improved in order to heighten the quality and rejection rate of the keywords. This needs to be done in order for either of the approaches to be a feasible keyword extraction technique used in the Swedish legal domain.

7.1 Future Work

Continued work could be adding more linguistic features to heighten the quality and lower the rejection rate of keywords. It would also be good to make a lager evaluation which could yield more accurate results. For this study there was not enough time to work on getting more participants and also participants who had

legal knowledge.

Future work could also include getting data which have been pre-tagged with manually extracted keywords by a legal expert which could yield another measure for the quality, and quantity, of the keywords.

If the points mentioned above were improved then the work of finding methods which could predict rulings based on previous rulings and their grounds could be done.

Bibliography

[1] Yukio Ohsawa, Nels E Benson, and Masahiko Yachida. Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *Research and Technology Advances in Digital Libraries, 1998. ADL 98. Proceedings. IEEE International Forum on*, pages 12–18. IEEE, 1998.

[2] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[3] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[4] Eibe Frank, Gordon W Paynter, Ian H Witten, Carl Gutwin, and Craig G Nevill-Manning. Domain-specific keyphrase extraction. In *IJCAI*, volume 99, pages 668–673, 1999.

[5] Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley. Automatic text structuring and summarization. *Information Processing & Management*, 33(2):193–207, 1997.

[6] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2(3):258–268, 2010.

[7] Eduard Hovy and Chin-Yew Lin. Automated text summarization and the summarist system. In *Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998*, pages 197–214. Association for Computational Linguistics, 1998.

[8] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[9] Caroline Uyttendaele, Marie-Francine Moens, and Jos Dumortier. Salomon: automatic abstracting of legal cases for effective access to court decisions. *Artificial Intelligence and Law*, 6(1):59–79, 1998.

[10] Tommaso Agnoloni, Lorenzo Bacci, and Maria Teresa Sagri. Legal keyword extraction and decision categorization: a case study on Italian civil case law. In *Semantic Processing of Legal Texts (SPLeT-2014) Workshop Programme*, page 1.

[11] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009.

[12] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, et al. *Introduction to data mining*, volume 1. Pearson Addison Wesley Boston, 2006.

[13] Kuo Zhang, Hui Xu, Jie Tang, and Juanzi Li. Keyword extraction using support vector machine. In *Advances in Web-Age Information Management*, pages 85–96. Springer, 2006.

[14] Hassan Sayyadi and Louiqa Raschid. A graph analytical approach for topic detection. *ACM Transactions on Internet Technology (TOIT)*, 13(2):4, 2013.

[15] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing*, volume 12, pages 44–49. Citeseer, 1994.

[16] Erik F Tjong Kim Sang and Sabine Buchholz. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, pages 127–132. Association for Computational Linguistics, 2000.

[17] Dipak L Chaudhari, Om P Damani, and Srivatsan Laxman. Lexical co-occurrence, statistical significance, and word association. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1058–1068. Association for Computational Linguistics, 2011.

[18] Paul R Kroeger. *Analyzing grammar: An introduction*. Cambridge University Press, 2005.

[19] Anand Rajaraman, Jeffrey D Ullman, Jeffrey David Ullman, and Jeffrey David Ullman. *Mining of massive datasets*, volume 1. Cambridge University Press Cambridge, 2012.

[20] Martin F Porter. Snowball: A language for stemming algorithms, 2001.

TRITA -EECS-EX-2020:50