ENGLISH
SUMMARIZATION OF ICELANDIC TEXTS

Karin Christiansen

Master of Science
Computer Science
June 2014
School of Computer Science
Reykjavík University

M.Sc. PROJECT REPORT

ISSN 1670-8539

Summarization of Icelandic Texts

by

Karin Christiansen

Project report submitted to the School of Computer Science
at Reykjavík University in partial fulfillment of
the requirements for the degree of
Master of Science in **Computer Science**

June 2014

Project Report Committee:

Hrafn Loftson, Supervisor
Associate Professor, Reykjavík University

Hannes Högni Vilhjálmsson
Associate Professor, Reykjavík University

Marta Kristín Lárusdóttir
Assistant Professor, Reykjavík University

Summarization of Icelandic Texts

Karin Christiansen

June 2014

Abstract

The field of text summarization has been evolving along with advances in Natural Language Processing (NLP) and Computer Science but until now no known attempts have been made to create a summarizer for summarizing Icelandic texts. This project looks into the field of text summarization, giving an overview of different types and approaches towards automatically generating summaries. Additionally, a clarification towards the requirements of language resources and scope towards implementing a summarizer for Icelandic texts is given.

Two known methods for text summarization (*TextRank* and *TFxIDF*) were selected for a prototype implementation, both methods rely on repetitions within the text to calculate a sentence score for identifying extract worthy sentences to include in a summary. The methods are both unsupervised and proved to be a good for a base summarizer to tailor for Icelandic texts. The methods were tested on a set of news articles. The quality of the summaries generated by the summarizer were evaluated by comparing them to human created summaries, the evaluation showed that the summarizer performs consistently and produces readable summaries.

Íslenskt heiti
Samantekt á íslenskum texta

Karin Christiansen

June 2014

Útdráttur

Acknowledgements

First of all I would like to thank my supervisor Hrafn Loftson for excellent guidance during the course of this project. He has always been available to answer questions and his feedback for this project has been invaluable. He has been very helpful and providing me with some of the datasets needed to successfully implement parts of my solution. Thank you very much Hrafn and thank you for encouraging me to take on this project.

I would like to thank Stella Guðjónsdóttir and Katrín María Víðisdóttir for volunteering for the tedious task of manually creating summaries for the evaluation dataset. Your contribution and patience has been invaluable and I owe you many thanks.

Many thanks to Shishir Patel for words of encouragement, support and taking the time to help proof reading this report. Also, I would like to thank the MSc. committee for their constructive, in-depth comments and feedback.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Reading lengthy text can be tedious and time consuming and sometimes this is a task many of us would rather not be forced to undertake, but what if an accurate summary of the full text existed and we could be confident that this summary accurately represented the information of the full text. This would free up time and resources, giving the individual time to focus on other things. However, generating accurate informative summaries is not trivial. There are several considerations to take into account, some of which relate to the requested summary output format or to the choice of method for generating the summary. The language of which the full text is written in can also affect the quality of the generated summary, since summarizers tend to be created for a specific language or set of languages. Currently there exists no summarizer tailored for generating summaries from Icelandic texts[1], this presents us with one of the main challenges of this project, namely to investigate what is required and can be done to implement a summarizer for Icelandic texts.

Additionally, it is necessary to get an understanding of what a summary is and what its core components consist of. The two main requirements to a good quality summary are: First, it should be concise and short, the summary should be a shorter version, being no more than 50%, of the original text. Second, a summary should channel the important information of a given text with the main goal of presenting the central core content (D. R. Radev, Hovy, & McKeown, 2002). Studies have shown that when dividing texts into important and less important parts the information content of a document appears in bursts, making it possible to form a distinction between the relative importance of a phrase (D. R. Radev et al., 2002). This distinction between what to identify as important and

[1] To the best knowledge of this author.

informative and what to exclude is one of the main challenges when generating automatic summaries.

1.1 Text Summarization

One of the early text summarization methods was proposed by Luhn (Luhn, 1958). His approach was to use frequency count of words and phrases to identify important content for generating abstracts of technical papers. This approach is naive but gives a useful angle on how summaries can be generated from any given text. Later, as the areas of Natural Language Processing (NLP) and Machine Learning have become more and more evolved, more advanced methods for text summarization were proposed. Some of these methods are briefly outlined in Chapter 2.

When developing methods for creating text summaries there are different approaches to use. One option is trying to cover a broad spectrum by developing a method to generate summaries for multi-language and genre independent texts. On the other hand, summaries can be generated by making use of language and domain specific details like part-of-speech (POS) tagging, detection of synonyms and using knowledge about the structure of the specific text. This does however require that the proper NLP tools and Thesauruses are available, along with the needed collection of corpora.

For Icelandic texts, the development of the IceNLP toolkit (Loftsson & Rögnvaldsson, 2007) has made it possible to use POS-tagging and lemmatization especially designed for the Icelandic language. This gives the possibility of tailoring a summarizer tool to Icelandic texts. The benefits of this language toolkit becomes apparent when syntactic filters are needed for the extraction of sentences for summarization. Even though natural languages are different and the IceNLP toolkit has been designed especially for the Icelandic language, it is possible to use other more general POS-taggers for the same tagging. The Standford POS-tagger [2] (Toutanova, Klein, Manning, & Singer, 2003) is an example of a POS-tagger that can be trained to tag texts written in Icelandic.

1.2 Possible Usage of a Summarizer

The potential usages for a summarizer customized for the Icelandic language are many, like for example, creating summaries for newspaper articles. These could be presented to

[2] http://nlp.stanford.edu/software/tagger.shtml

the reader, such that a quick overview of the daily news feed is given in a time efficient way. Especially, since many papers update their news websites several times a day but for readers there is a limited time and interest to read all articles in depth.

Other areas of usage could be within businesses where it is required to get a quick overview of a lengthy text spanning several tens of pages. A concise summary could speed up decision making processes and ease the workload of the reader.

1.3 Methods

For this project, two methods for text summarization have been implemented, plus a simple baseline method. The first method is *TextRank* (Mihalcea & Tarau, 2004), a graph- and centrality-based method that uses co-occurrences in the text to identify similarities between sentences and uses *PageRank* (Page, Brin, Motwani, & Winograd, 1999) for determining the ranking score of each sentence. The second method is the Term Frequency-Inverse Document Frequency (*TFxIDF*), which is frequency based and uses the frequency count of a word in a text and in a background corpora to determine the word's relative importance in a given text versus the word's significance in the overall language. The basic idea is, that if the frequency of the word is high in a text but the overall frequency of the word in the language is low, then this word is considered to be an important term.

A simple baseline summarizer was also implemented to use in the evaluation process as a benchmark against the two other implemented methods. The baseline summarization method was implemented in such a way that it was consistent with the baseline method used in the *TextRank* paper.

1.4 Goals

What are the current existing approaches and methods in text summarization and are there any methods that can be used to develop a summarizer tailored for generating summaries from Icelandic texts?

To answer this question the goal is to look into current methods applied for text summarization and from this get an overview of existing approaches that can be used for implementing a prototype summarizer for Icelandic texts. A solid baseline summarizer must also be implemented for comparing to the other implemented methods, this way the methods/summarizer will have a quality threshold to beat. Furthermore, it is important that the generated summaries can be evaluated accordantly to recognized standards within the field of text summarization, such that their quality can be assessed and compared to other published results. The idea is that a comparison of automatically generated summaries to human created summaries will give an immediate quality assessment of the summaries generated by the prototype summarizer.

1.5 Overview of Report

This report is structured as follows: Chapter 1 gives an introduction to the topic of text summarization, the outline of this report and the goals of the project. Chapter 2 goes into the topic of text summarization explaining some concepts and approaches for text summarization and gives an overview of some known methods used for automatic text summarization. Chapter 3 describes in detail the methods that are the basis of the implemented text summarizers created for this project. Chapter 4 describes the experiments conducted and lists the results obtained. Chapter 5 gives a brief outline of future work for this project. Chapter 6 gives the final conclusion. Appendix A lists some example summaries generated by the summarization methods implemented for this project along with the human generated summaries for the same texts. Appendix B shows the GUI of the summarizer and gives a brief introduction to its use.

Chapter 2

Text Summarization

The area of text summarization has been evolving for decades, starting in the early days of computing by Luhn (1958) (see Section 1.1) and as the fields of NLP and Computer Science have developed, many techniques from these two inseparable areas have been employed to improve the quality of automatically generated summaries. In this chapter a brief overview is given of the area of text summarization. This includes a description of types of summaries, along with some general introductions of methods and systems to generate automatic summaries.

2.1 Types of Summaries

Text summarization can be divided into three main types. These depend on what the source of the summary is, what approach is used for generating the summary and what the final end result of the summary is.

1. The final summary is an abstract or collection of extracts.
2. The summary is generated from a single document or multiple documents.
3. The summary is created by taking a generic or query-based approach.

2.1.1 Summary Outcome

The are two general outcome types of summaries, namely extractive and abstractive. The extractive summary is the simpler kind of summary as it only extracts sentences and/or phrases from the full text to generate the summary. Abstractive summaries on the other hand use different words to describe the content of the full text when creating the summary. Most current automatic summarizers are extractive, since generating an extractive summary is simpler than generating an abstractive summary (Jurafsky & Martin, 2009). The creation of an abstractive summarizer sets high demands for the post-processing of the summary, as rearranging sentences may be necessary along with the removal of unnecessary phrases and use of synonyms to make the generated summary more readable and coherent. The extractive summarizer on the other hand gives only a selection of sentences and phrases from the original text. This can give a lesser impression of readability but this does however not affect the validity of the summary as it gives a clear impression of which parts of the text are more representative than others.

2.1.2 Summary Sources

When creating summaries, a distinction is made between single document summarization and multi-document summarization. When generating single document summaries, the summary is generated from a single original document. In contrast, multi-document summaries are generated from several documents. Multi-document summaries are harder to implement (Jurafsky & Martin, 2009) but seem to perform better as repetition across documents can be used to indicate the significance of a sentence or phrase. Single document summaries, on the other hand, seem to be easier for humans to generate (Nenkova, 2005) as there is a clearer scope and overview of the text at hand. Since summaries generated by humans are needed for quality comparisons with the automatically generated summaries and the reference summaries created for quality comparisons for this project are created by two volunteers. The focus on single document summarization will speed up the process of creating these reference summaries since, as stated before, it is easier for humans to generate single document summaries. This means that the summarizers implemented here only handle single document summarization as the scope of this project is limited but further future extensions can be made to handle multi-document summarization.

2.1.3 Summarization Approaches

Summaries are further categorized as either generic or query-based. Generic summaries do not consider any particular end user or information needed for the final summary. Query-based summaries or topic-based summaries are generated as an answer to a user query (Jurafsky & Martin, 2009). In topic based summaries, sentences that contain certain topic or query words are given higher rankings in order to give higher probability for extraction selection (D. R. Radev et al., 2002).

The summaries generated in this project are generic summaries as the summarizers simply returns the highest ranking sentences for the summary.

2.2 Single Document Summarization Overview

Extractive summarization can be divided into three main tasks (Jurafsky & Martin, 2009):

1. Content Selection, where the relevant sentences to extract are chosen.
2. Information Ordering, where the ordering of the extracted sentences is determined.
3. Sentence Realization, where clean up is done on the sentences, for example by removing non essential phrases.

2.2.1 Content Selection

Content selection can be divided into two techniques, namely unsupervised and supervised. The unsupervised content selection methods use frequency count and similarity measurement of occurrences within the text to identify sentences to select for the summary. Supervised content selection relies on background corpora, often using positioning and machine learning techniques along with frequency and similarity measurements to identify which sentences to include in a summary.

Unsupervised Content Selection

Early methods of text summarization included unsupervised approaches like word frequency count in a document (Luhn, 1958). Luhns approach was to compute the most frequent words in a document and from this identify the most extract worthy sentences in order to generate a summary. This approach does however present a problem as a word may have a high frequency in a language but may not be important to the topic of the text to be summarized. To overcome these issues, weightings methods like *TFxIDF* and *log-likelihood ratio* can be used. Both these methods use statistics along with normalization to compute the final importance of a word in a given document. The score of a sentence, can be calculated by summing the weights of the words in the sentence.

TFxIDF counts term frequency in a document and in a background corpus, to determine the overall frequency salience of a word in a document. By using a background corpus *TFxIDF* can give words that have a high frequency in the language a lower score. This way words that have a relative high frequency in the document but not in the overall language will get rated as more extract worthy than other frequent words. Formally, this is set up as Equation 2.1, where $TF_{i,j}$ is the term frequency of the term i, and j is the document containing the term, whose frequency is to be counted. IDF_i is the inverted document frequency, telling us if the term is frequent in a collection of documents. The IDF_i is calculated by dividing total number of documents with the number of documents in the background corpus that contain the term.

$$weight(w_i) = TF_{i,j} \; * \; IDF_i \tag{2.1}$$

The scores are summed for each sentence such that sentences containing more salient words are selected for extraction, see Section 3.2 for more detailed description.

The *log-likelihood ratio*, or for short, *LLR* is a simple unsupervised method for text summarization. This method does, like the *TFxIDF*, use the probability of observing a word in a document and in a background corpus as a measure of how important a given word is but unlike *TFxIDF*, *LLR* sets a threshold to determine if a term is descriptive or not. If a word is determined to be descriptive it is assigned the weight 1, otherwise it is assigned the weight 0. The score for each sentence is defined as the average weight of its terms. The highest scoring sentences are then included in the summary (Jurafsky & Martin, 2009).

$$weight(w_i) = \begin{cases} 1 & \text{if } -2log(\lambda(w_i)) > 10 \\ 0 & \text{if } otherwise \end{cases} \tag{2.2}$$

The *LLR* and *TFxIDF* belong to the centroid-based algorithms. These are a category of algorithms where the sentences containing the most informative words act as clusters and the centroid is the most important sentence, the goal is to find the sentences closest to this centroid.

Another category of algorithms useful in text summarization are the centrality-based algorithms. These differ from the centroid-based by also taking into account how central the individual sentences are within the text. Rather than just computing the weighted frequency of the individual words within the document, centrality-based systems take into account the similarity of the sentences. This way an average score is calculated for the importance of each sentence based on the relative similarity to other sentences. The co-occurrence within the sentences can be shared words, shared word classes, shared n-grams, longest common subsequence or other lexical information derived from a lexical resource, like for example WordNet[1] which is a large lexical database of the English language.

The similarity measurement can, if for example based on frequency-based occurrences, be calculated as the *Cosine Similarity* between two sentences. The *Cosine Similarity* with *TFxIDF* weights is one of the often used methods for similarity measurement (Nenkova & McKeown, 2012). The sentences are represented as bag-of-words vectors where the weights are calculated for the terms in the sentence. The two sentences can be described as the vectors $\vec{S_i}$ and $\vec{S_j}$, where $weight(w)$ is the weighting ratio from Equation 2.1 but can be replaced by any other weighting formula appropriate.

$$\begin{aligned}\vec{S_i} &= (weight(w_{i_1}), weight(w_{i_2}), weight(w_{i_3}), ...weight(w_{i_n}));\\ \vec{S_j} &= (weight(w_{j_1}), weight(w_{j_2}), weight(w_{j_3}), ...weight(w_{j_n}));\end{aligned} \tag{2.3}$$

The similarity between the vectors is then calculated by taking the product of $\vec{S_i}$ and $\vec{S_j}$ and divide it with the product of the Euclidean length of the two vectors.

$$simmilarity(\vec{S_i}, \vec{S_j}) = \frac{\vec{S_i} \cdot \vec{S_j}}{||\vec{S_i}|| \cdot ||\vec{S_j}||} \tag{2.4}$$

The score of any given sentence is then the average cosine distance to all the other sentences in the text (Jurafsky & Martin, 2009). Using the co-occurrence similarity measurement approach, the sentences most similar to all other sentences in the text are extracted for the summary. The core of centrality-based methods is that important information is repeated in some sentences across the text and this way it is possible to determine which sentences are the most informative in the text.

TextRank (Mihalcea & Tarau, 2004) is an example of a summarization algorithm that is centrality-based. The *TextRank* method uses a syntactical filter to identify the centrality similarities between the sentences and uses *PageRank* to rank them accordantly to their score. *TextRank* is described in more detail in Chapter 3.

[1] `http://wordnet.princeton.edu/`

Supervised Content Selection

When applying supervised content selection options like the positioning of the sentences in the text are taken into account. This positioning approach is especially useful for generating summaries for specific types of texts, like newspaper articles (Nenkova, 2005) and this approach is used in for example *SweSum* (Dalianis, 2000). When generating summaries of technical documents, Saggio & Lapalme (2002) found by comparing to human generated summaries that the extract worthy sections of a text were distributed among the abstract, the first section, headlines and captions of the text.

Other text characteristics that can be taken into account are sentence lengths, where sentences shorter that some threshold are ignored. Cue phrases from a predefined list can also give indication of sentences that are extract worthy (Nenkova & McKeown, 2012). These cue phrases can be in the form of "in conclusion", "this paper presents", etc. (Edmundson, 1969; Hovy & Lin, 1998). The usage of cue phrases is however hard to make context independent since different types of texts have different cue phrases (Barzilay & Elhadad, 1997).

By using Supervised Machine Learning it is possible to combine all of the above mentioned techniques and use them as features in a classifier (Jurafsky & Martin, 2009). A corpus of document and summary pairs, where the summaries have been created by humans, is used as the training basis for which the features are probable of creating the best summaries. Sentences in the training documents are then labelled depending on whether they appear in the training set summaries or not. This method is useful for creating extractive summaries since it is easy to identify whether sentences are included in the training set or not. However, the task of creating the corpus of document and summary pairs is time consuming and the content must be diverse in order to not make the corpus domain dependent. The Ziff-Davis corpus (Marcu, 1999) is an example of a corpus of document and summary pairs in English. Currently no such corpus is available for Icelandic texts.

External Resources for Content Selection

A useful feature selection for sentences are the lexical structures within it, like common linked elements in the text. WordNet is well suited for the feature selection task as it groups nouns, verbs, adjectives and adverbs into sets of synonyms called synsets. The synsets provide short and general definitions of words/phrases and records the various semantic relations between the synonym sets. The use of WordNet for finding lexical chains and similar meanings of phrases containing for example synonyms is a solid method for finding extract worthy sentences for summary generation. However, currently there is no Icelandic equal to WordNet. There exists the IceWordNet[2] but in its current form it is not any where near as extensive as WordNet. IceWordNet contains 5,000 words while WordNet contains around 155,300 words. The Summarist (Hovy & Lin, 1998) is a text summarizer that uses statistical features and information retrieval but it further relies on topic identification by linking to resources like WordNet to determine a word's frequency. The idea is that the more frequent a topic related word is the more important it is, for example the words *seat*, *handle*, *wheel* all relate to a bicycle. This way Summarist can generate the summary not only from single word frequency but also related words/topics and synonyms.

Wikipedia[3] is another vast resource that either on its own, or in combination with other resources can be used for topic identification and relations, when generating summaries. This approach has been used in topic-driven summarization (Nastase, 2008; Nastase, Milne, & Filippova, 2009) where Wikipedia was used to identify concepts within the documents to be summarized. Scores were then given according to their significance and a machine learned confidence, determining, if they should be included in the summary. On Wikipedia there are currently (April 2014) over 37.000 articles listed as written in Icelandic. Although this number is significantly lower than the number of articles written in English (close to 4,5 millions) this seems to be an interesting approach, considering the fact that pages on Wikipedia cover a wide spectrum of topics and areas, all the way from ancient history to current day popstars.

2.3 Multi-Document Summarization Overview

In multi-document summarization a single summary is generated from several original documents. This approach is well suited for generating summaries from web-sources, like news sites and blog sites, where the content is event and/or topic based. Multi-document summarization follows the same steps as single document summarization: Content Selection, Information Ordering and Sentence Realisation. The methods and resources described in Section 2.2.1 are also applicable to multi-document summarization. But the challenge for multi-document summarization lies in generating a coherent summary from many perhaps repetitive sources. However, this repetitiveness is also one of the strengths of multi-document summarization. By clustering candidate sentences from all the docu-

[2] `http://www.malfong.is/index.php?lang=en&pg=icewordnet`
[3] `https://www.wikipedia.org/`

ments together and penalizing repeating sentences in the cluster it is possible to identify which sentences are the most appropriate for inclusion in the final summary (Jurafsky & Martin, 2009).

2.4 Information Ordering

The challenge in information ordering or sentence ordering is to arrange the extracted sentences in such a way that they are as coherent as possible. In some cases, often in single-document summarization, the natural ordering of the sentences in a text can be used as the sentence ordering. This approach does however, not grantee that the summary automatically will become the most readable. Another simple method of arranging the sentence order is to use the ranking ordering of extracted sentences. Using this ordering, sentences that stick out in the text as the most important and informative will appear in the beginning of the summary.

When dealing with multi-document summarization the challenges of sentence ordering become even more apparent since sentences come from multiple sources. Using the natural ordering of the sentences would possible not give the most readable result as the information content of the documents could be presented in different order from document to document. The chronological ordering approach takes into account the novelty of the documents from which the sentences are extracted and orders the sentences by the dating of the documents. This approach does though, not solve the issue of the information ordering but only assumes that the newest content should appear first. Majority ordering is a different ordering approach where similarity of the ordering across the documents is used to determine the sentence ordering in the summary (McKeown, Klavans, Hatzivassiloglou, Barzilay, & Eskin, 1999). This approach does however fall short if the original ordering of the sentences is very different for each document, as then there is no majority order to follow. Topic ordering is an ordering method that tries to order the sentences by what topic they describe. The sentences are extracted from their original documents and clustered into topic clusters such that the information in the summary is presenting one topic at the time to the reader. In (Barzilay, Elhadad, & McKeown, 2002) this approach is used in combination with chronological ordering to determine the order of the topic clusters.

2.5 Sentence Realisation

After a summary has been generated from an original text, it is possible that some of the extracted sentences can be compressed for simplification. This can increase the conciseness and readability by removing unnecessary redundant and irrelevant information. By examining the grammar of the sentences, it is possible to identify some grammatical rules that can be used to compress the sentences. This simplification compression can include removing phrases that start with the wordings "For example", "As a matter of fact" as these are all identified as initial adverbs. Other grammatical distinctions like appositives and attributive clauses are also used for compressing sentences (Jurafsky & Martin, 2009). A few simplification rules applicable to sentence simplification are:

- **Appositives**: John and Paul, ~~both friends of mine~~, are starting a band.
- **Attributive clauses**: The first single is released in May, ~~Paul exclaimed proudly~~.
- **Initial adverbs**: ~~However~~, they are not rich and famous yet.

Other approaches have also proven useful for sentence simplification. Lin and Hovy (1998) used the entity hierarchy of WordNet to identify phrases for simplification. They used concept generalization to convert phrases like "John bought some apples, pears, and oranges" into "John bought some fruit".

2.6 Baselines

In order to evaluate an automatically generated summary a baseline summary can be used for comparison. The baseline summary can be generated in a number of ways depending on the nature of the text or the method of the summarizer, it is to be compared to. An example of a typical method for generating baseline summaries is the *Lead-based approach*, where the first n sentences in a document are selected for the summary. Another approach is to select sentences at random from a document, to include in the baseline summary. Other approaches include using a positional feature for generating the baseline summary, like for example selecting the headline and the first sentence of the second paragraph (Das & Martins, 2007).

MEAD (D. Radev et al., 2004) is an extractive summarizer framework that can be used for both single- and multi-document summarization. MEAD implements multiple summarization algorithms, including lead-based and random baseline summarizers. However, when generating baseline summaries one is not restricted to lead-based and random baseline summaries. In MEAD the default scoring of sentences is based on three elements: sentence length, centroid score and position in text. These are all configurable, making it possible to generate more customized baseline summaries.

2.7 The DUC and TAC Conferences

The Document Understanding Conferences (DUC)[4] and Text Analysis Conference (TAC)[5] support the NLP community by encouraging research based on large common test collections. The DUC conferences ran from 2001 until 2007 focusing on efforts in the field of text summarization. The goal was to enable researchers to participate in large scale experiments by sending out sets of evaluation and training data. The challenges related to the test sets and changed from year to year, from summarizing and extracting keywords from newspaper articles to cross-lingual multi-document summarization. In 2008, DUC became the text summarization track of the TAC conference running as a individual track until 2011. The *TextRank* paper (Mihalcea & Tarau, 2004), which describes one of the methods implemented for this project (see Chapter 3) uses the dataset from DUC 2002 as basis of its evaluation. The DUC 2002 dataset consists of 567 news articles. One of the three tasks presented for the participants in 2002 was to generate a 100 word generic-summary from a single document and this is also the task the paper does address.

[4] http://www-nlpir.nist.gov/projects/duc/
[5] http://www.nist.gov/tac/

Chapter 3

Methods Used

When implementing a summarizer, a number of relevant choices must be made. Some of these where discussed in Chapter 2. It is important that the implemented method is suitable for the Icelandic language, meaning that it given a text written in Icelandic has the best prerequisites to generate a good summary. *TextRank* (Mihalcea & Tarau, 2004) and *TFxIDF* are examples of methods that can be implemented for the Icelandic language. *TFxIDF* can be implemented to be completely language independent while *TextRank* needs some language specific components in the form of a POS-tagger. The most fundamental difference between these two methods is the basic structure. *TextRank* is graph-based and depends on co-occurrence to find extract worthy sentences while *TFxIDF* is frequency-based and uses frequency count to find the most extract worthy sentences.

Two versions of the *TextRank* algorithm were introduced by Michalea & Taru (2004), one for keyword extraction and the other for sentence extraction. The *TextRank* method is described in detail in the paper (Mihalcea & Tarau, 2004). The keyword extraction part is available online as open source code implemented for English and Spanish texts[1]. In this project this open source code is used as basis and was extended to also extract keywords from Icelandic texts. The code was then further extended to generate summaries from Icelandic texts. However, it is worth pointing out that since the focus of this project is summarization of Icelandic texts, the original code has been simplified to only include parts useful for this task. The code for the methods implemented in this chapter can be found at `https://github.com/karchr/icetextsum`.

[1] `https://github.com/samxhuan/textrank`

3.1 TextRank

TextRank is an unsupervised graph-based method for keyword- and sentence extraction. *TextRank* relies on the co-occurrence between words of specific word-classes in sentences. This co-occurrence is used to rank the sentences according to importance and information content against all the other sentences in the text.

TextRank uses the principles of *PageRank* (Page et al., 1999) to rank the most important keywords/sentences in the text. Each node in the graph is a text unit, either a single word or a whole sentence depending on if keyword or sentence extraction is being performed. The co-occurrence between the nodes gives a recommendation voting between the nodes such that the nodes with a higher number of incoming edges get a higher rank. *PageRank* is designed to identify the relative importance of a web-page based on its in- and outgoing links and is the main core of Google's search engine. The basic idea is that more often a page has been linked to the more important it is, since removing a page with many links will affect the content graph more than removing a page with few links. This link counting system is also described as the *Recommender* or *Voter System*. The importance of a recommendation vote is higher, if it comes from a node with a high ranking because in *PageRank* the scoring weights of the nodes of the incoming edges is taken into account when calculating the score of the current node. This way, the score of a node is always relative to the other nodes in the graph.

Formula 3.1 shows the *PageRank* algorithm, where $S(V_i)$ is the score of the current node V_i and $S(V_j)$ is the score of the previous node V_j. $j \in In(V_i)$ are the edges that go from V_j to V_i and $Out(V_j)$ is the number of outgoing edges from V_j. The score and the count of outgoing edges of the node V_j will have a direct influence on the score of the node V_i and by applying this process to the whole graph the relative importance of the nodes becomes apparent and a ranked hierarchy is established. The *PageRank* algorithm uses the damping factor d to take into account the probability of random events, also called the *Random Surfer Model*. This event is modelled by the damping factor d, which is set to a value between 0 and 1. The probability d of a surfer following a link on a page and the probability of the surfer jumping to a completely different page is $1 - d$. The damping factor d is usually set to the value 0.85 which also is the damping factor value used for *TextRank*.

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \tag{3.1}$$

The authors of *TextRank* use this same ranking principle to find the importance of the individual sentences in a text. They base the connecting page links on lexical co-occurrence between the sentences. The *TextRank* authors use a slightly modified version of the *PageRank* algorithm described in Equation 3.1. Since the connections between lexical elements can be described in a different way than the connections between web pages. See subsections 3.1.2 and 3.1.3 for more details.

Although this implementation of *TextRank* and the one described in the *TextRank* paper uses *PageRank* for finding the most extract worthy sentences in a document, other graph-based ranking algorithms like the *HITS* algorithm (Kleinberg, 1999) can be used just as well. An approach to extract keywords from a document text using the *HITS* algorithm can be found in (Litvak & Last, 2008).

One of the clear advantages of *TextRank* is that it is fully unsupervised and does not require any background corpus of already compiled summaries to work. Therefore, it is easy to get started with the implementation as no effort has to be put into creating a training set of summaries, like it is required for supervised methods. Furthermore, *TextRank* doesn't require any collection of documents to calculate probabilistic frequency from, like *TFxIDF* does.

3.1.1 Main Processing Steps

In this project, the process of identifying and extracting the most important elements in a given text consists of five main steps.

1. Tokenization, splitting the text into sentences and individual tokens.
2. POS-tagging, marking each token with its POS-tag.
3. Graph construction, a node consists of a keyword or a sentence.
4. Calculating keyword/sentence nodes weights and rank.
5. Sorting by rank.

The tokenization and POS-tagging are done using the *IceNLP toolkit* (Loftsson & Rögnvaldsson, 2007). This toolkit has been developed especially to handle the Icelandic language and its morphological nature. After the tokenization and POS-tagging, all the words not categorized as nouns, proper nouns, numerals or adjectives are filtered out. By limiting the number of included word classes, keywords will become more easily identifiable as we typically think of these as nouns modified by adjectives. The original *TextRank* only considers nouns and adjectives, as this was shown to give them the best results. In this project, the included word classes have been extended to numerals as this has shown to give good summary results.

Once the irrelevant words are filtered out the *TextRank* graph is constructed. The graph nodes are created with the following variables: *edges*, *rank*, *weight*, *value*, *key* and *marked*. The *key* variable is used to identify the node in the graph and the *marked* variable is used for marking nodes in the n-gram construction for keyword extraction (see Section 3.1.2). The *edges* variable holds the nodes edges, the *rank* variable stores the nodes ranking score and the *weight* variable holds the nodes weighting score. The *value* variable is a holder for the nodes plain text value, which is either a keyword or a whole sentence.

Node	
- key:	int
- edges:	<Set>
- rank:	double
- weight:	double
- marked:	boolean
- value:	NodeValue

Figure 3.1: *TextRank* graph node.

In the final steps, the weights and ranks of each node are calculated. These are calculated according to the basic principles of the *PageRank* formula shown in 3.1, until the graph converges. The convergence is achieved when the error rate for any node in the graph falls below a threshold of 0.0001. The error rate of a node is defined by the difference between the weighting score $S^{k+1}(V_i)$ and the weighting score $S^k(V_i)$ of the node computed at the current iteration k. Since there is no way of knowing the real score before hand, the error rate is approximated as the difference between the weighting scores computed at two successive iterations $S^{k+1}(V_i) - S^k(V_i)$. The error rate approximation is then estimated from the *standard error of the mean*: $\sigma_{\bar{X}}$ for every node at every iteration, until graph convergence.

3.1.2 Keyword Extraction

The keyword extraction starts with tokenization, POS-tagging and filtering as described above. Only words belonging to a predefined set of word classes are added to the graph as potential keywords. A filter allowing only nouns, proper nouns, adjectives and numerals to be added to the graph has shown to give the most reasonable results. However, no proper experiments have been done to prove this claim. By proper experiments, it is referred to an experiment where comparisons to humanly assigned keywords would be compared to the automatically extracted keywords to make any final conclusions about which filter combination would be the best one. This would be the proper way to evaluate the quality of the *TextRank* keyword extraction algorithm modified to extract keywords from Icelandic texts.

Basically, the detection of keywords is split into the following six processing steps. These steps should be seen as the more detailed versions of the steps 3 to 5 listed in Section 3.1.1.

1. **Lexical Filter and Graph Construction**
   After the tagging of the full text, the text is split into sentences and added to a sentence holder as a collection of tokens. These tokens are then filtered for any irrelevant word classes such that only nouns, proper nouns, numeral and adjectives are seen as potential keywords. These keywords are then added to the graph as node objects connected by undirected edges.

2. **First TextRank Run**
   The second step is to run the first iteration of TextRank to determine the ranking score of the added tokens. After this the top ranking keywords are marked for further processing for n-gram detection, in this context a n-gram is a token. The n-gram detection is done by checking if any found keyword is directly adjacent to another keyword.

3. **N-gram detection**
   The top ranking nodes are checked to see if they are part of a n-gram. If this is the case the keyword and the directly connected words are merged into a n-gram node and added to the new $ngram\_subgraph$ graph.

4. **Add the N-gram sub-graph to the TextRank graph**
   In the fourth step the newly created $ngram\_subgraph$ is connected to the full TextRank graph, resulting in a merged graph.

5. **Run TextRank on the merged graph(Second TextRank Run)**
   This step is actually a repetition of the second step, now performed on the merged graph.

6. **Sort Keywords by Rank**
   Found keywords are added to a list and sorted by highest to lowest rank.

Each step is listed briefly above, but is explained in more detail in the following text. The six steps are identified in the explanatory text as (Step 1), (Step 2), etc.

During the graph construction (Step 1) undirected edges are created between words that co-occur in a window of N words. For this project $N = 2$ was used. The use of undirected edges differs from the traditional *PageRank* algorithm where directed edges are used. The ranking of the nodes (Step 2) is calculated by running the modified *PageRank* algorithm shown in Equation 3.2. Initially all ranks of the nodes are set to 1 and as the rank calculation iterates through the graph the ranks are updated to their approximated values until the graph converges.

$$S(V_i) = (1 - d) + d * \sum_{j \in V_i} \frac{1}{|j \in V_j|} S(V_j) \tag{3.2}$$

The connections between the vertices in the *TextRank* graph do not take into account the literal meaning of the keywords. Two words are only connected if the underlying word class of the words are accepted by the filter. This means that the search for co-occurrence is done only by looking for words that fit the predefined filter within a window of two words. This means that for example, if two nouns are identified then an edge is created between these two nouns. All the other words not recognized by the syntax filter are set to null and are not connected to the graph. Figure 3.2 shows the connections between the words identified as potential keywords in an example sentence.

Veðurstofa Íslands spáir norðanhríð á föstudag
(The Icelandic Meteorological Office predicts snowstorm on Friday)

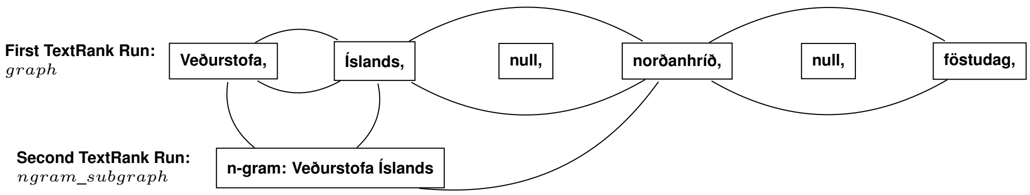


Figure 3.2: *TextRank* graph of keywords.

Figure 3.2 shows the merged graph after (Step 4). After the first converging of the graph, shown as "*First TextRank Run*" in Figure 3.2, the top ranking vertices are post-processed for connecting multi-word keywords, also referred to as n-grams in this context. This is done by constructing a new graph with compound keywords found by searching for n-grams in the text (Step 3). The n-gram detection is done by checking if any of the top

> *Bændur á tánum vegna illviðrisspár.*
>
> *"Ég held að menn geti ekki leyft sér að sitja bara heima þegar það spáir svona," segir Birgir H. Arason sauðfjárbóndi á Gullbrekku í Eyjafirði. Bændur um allt Norðurland eru á tánum vegna illviðrisspár um komandi helgi. Víða er fundað í kvöld og svo gæti farið að göngur hefjist sumstaðar strax á morgun.*
>
> *Veðurstofa Íslands spáir norðanhríð á föstudag með slyddu eða snjókomu í 150-250 metra hæð yfir sjávarmáli og vindhraða allt að 15-23 m/s. Á laugardagsmorgun er svo von á norðvestan 18-25 m/s á Norður- og Austurlandi og mikilli rigningu neðan við 100-200 metrum yfir sjávarmáli, en annars slyddu eða snjókomu.*

Figure 3.3: Extract from news article from mbl.is [2]

ranking keywords found in the first *TextRank* run are directly adjacent to another keyword. An example of a n-gram can be seen in Figure 3.2, where *Veðurstofa Íslands* is identified as a n-gram because these two keywords are directly adjacent in the text. The n-grams are added to a second graph, named $ngram\_subgraph$, this graph is then connected to the first main graph of keywords, as Figure 3.2 shows. New rankings are then calculated on this merged graph using the same ranking score calculations described by Equation 3.2 until the merged graph converges at the predefined threshold (Step 5). The ranking scores are then used to sort and produce a list of the most informative keywords (Step 6).

Keyword Extraction Example

The Keyword extraction calculates the number of extracted keywords from a percentage of the whole original text. Per default this ratio is set to a third of the number of nodes in the graph. This ratio is suitable for shorter texts like abstracts and the text in Figure 3.3. The text in Figure 3.3 is an extract of a news article from mbl.is, this text has for space reasons been shortened to only include the first two paragraphs.

Running the keyword selection on the text in Figure 3.3 returns the keywords:
Birgir H. Arason sauðfjárbóndi, 150-250 metra hæð, Veðurstofa Íslands, komandi helgi, mikilli rigningu, 18-25 m/s, 100-200 metrum, 15-23 m/s, göngur, Gullbrekku, menn, tánum, snjókomu, sjávarmáli.

The keywords extracted from Figure 3.3 show that the construction of n-grams gives a more complete and readable list of keywords. The compound keywords give a good con-

[2] `http:/www.mbl.is/frettir/innlent/2013/08/26/baendur_a_tanum_vegna_illvidrisspar`

text, since they give a further insight to the topic, unlike if they where listed as single one word keywords.

3.1.3 Sentence Extraction

The sentence extraction follows the five main steps listed in Section 3.1.1. After the tokenization, POS-tagging and filtering, the graph construction process starts with building a directed graph from the sentences where the edges represent the co-occurrences between the words in the sentences. Each node is weighted according to its number of in- and outgoing edges, similarity measurement to other nodes and the damping factor d, according to the *PageRank* algorithm principles.

The sentence extraction can be broken down to the below listed steps. These steps should be seen as a detailed version of the steps 3 to 5 listed in Section 3.1.1.

1. **Lexical Filter and Graph Construction**
   After the tagging of the full text, the text is split in to sentences and added to a sentence holder as a collection of tokens. The sentences are then added to the graph and connected with directed edges, going from currently selected node to the node it is being compared to.

2. **Run TextRank**
   In the second step the weighting of the sentences is calculated. Here each sentence is compared to all other sentences in the text and a similarity ratio is calculated (see Equation 3.3). The tokens in the sentences are filtered to identify interesting word classes such, that only nouns, proper nouns, numerals and adjectives are used in the similarity ratio calculations. The similarity measurement of two sentences i and j, w_{ij}, is used to calculate the weighting score for the given node, along with the score of the previous node and the damping factor d (see Equation 3.4). The ranking scores are run as iterations over the graph until the graph converges at a predefined threshold.

3. **Sort Keywords by Rank**
   The highest scoring sentences are added to a list and sorted by highest to lowest scoring rank.

In Step 2, the similarity between two sentences i and j is calculated by the count of matching words in sentence i and sentences j divided by the logarithm of the total length of the two sentences (see Equation 3.3). The latter is a sentence length normalizer to prevent the favouring of long sentences.

Like in the keyword extraction, only the underlying word classes of the words in the sentences are considered not the literal meaning of the words. Only nouns, proper nouns, adjectives and numerals are used in the similarity measurement. If any words of these four word classes are identified, a similarity is detected between the two sentences.

$$w_{ij} = similarity(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \; \& \; w_k \in Sj\}|}{log(|S_i|) + log(|S_j|)} \tag{3.3}$$

The similarity measurement $similarity(S_i, S_j)$, which also is written as w_{ij}, gives the ratio of the similarity between two sentences which then is used in the ranking calculations for each of the individual sentences in the text.

The rankings are calculated recursively through the graph taking the weight of the previous nodes into account. This way the rank of a node is always relative to the rank of all other nodes. Equation 3.4 shows the ranking calculation formula. The equation is an adapted version of *PageRank* shown in Equation 3.1 in Section 3.1. The same basic idea of taking into account the weights of other nodes(sentences) and calculating a similarity measure between the sentences, along with counting the number of in-going edges from V_j to V_i ($V_j \in In(V_i)$) and counting the number of outgoing edges from V_j to V_k ($V_k \in Out(v_j)$) is applied. The damping factor is used in the same way to take into account random jumps in the graph. A modification to the ranking formula in 3.4 is that instead of only dividing w_{ij} with the number of outgoing edges from V_j, w_{ij} is divided by the sum of the similarity measurements between outgoing edges from V_j to V_k. This way a similarity measurement is done involving three sentences.

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum\limits_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \tag{3.4}$$

The iterative process of calculating the weighting score WS of each node in the graph is shown by the labelling of nodes and edges in Figure 3.4. By looking at the figure we see that if the score WS of the node V_i is to be calculated, then by Equation 3.4, first the rank score of V_j and all other nodes that are connected to V_i by outgoing edges must be calculated first. But in order to calculate the ranking score of V_j the ranking score of all V_k nodes must be calculated first. This process continues through the graph and will finally result in the ranking score of V_i relative to the score of all the other nodes it is connected to, both the ones it connects to directly and the ones it connects to through other nodes.

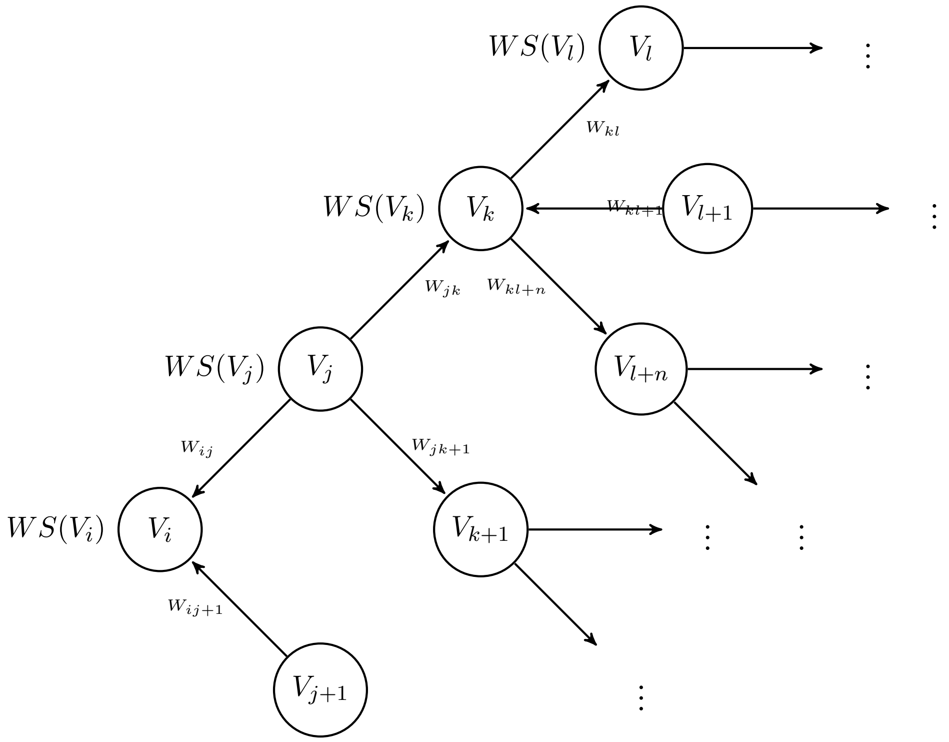


Figure 3.4: The *TextRank* Graph

To give an example, assume that the sentence S_i and sentence S_j have 2 matching words of the same word class and that the length of S_i is 25 words and the length of S_j is 12 words. By Figure 3.4 sentence S_i is node V_i and sentence S_j is node V_j. The following would be the similarity measurement ratio between the two sentences.

$$0.81 \approx \frac{2}{log(25) + log(12)}$$

Further, the similarity ratio w_{jk} and w_{jk+1} between the nodes V_J, V_k and V_{k+1} would be calculated in the same manner. The rank score would then be calculated according to Equation 3.4. Where $w_{jk} + w_{jk+1}$ represents the sum of the similarity measurements of the outgoing edges from node V_j in Figure 3.4.

$$WS(V_i) = (1 - d) + d * \frac{w_{ij}}{w_{jk} + w_{jk+1}} WS(V_j)$$

By assuming the ratio w_{jk}, w_{jk+1} and the ranking score of $WS(V_j)$ have already been calculated in a previous iteration, a possible scoring rank calculation would be the following:

$$0.48 = (1 - 0.85) + 0.85 * \frac{0.81}{0.80+0.29} * 0.52$$

These calculation steps would then be calculated for all the nodes until the convergence of the graph ranking scores was detected and the final ranking score would be returned. Figure 3.5 is an example of a *TextRank* graph with node and edge weights calculated. Initially all nodes have the weighting rank score of 1 but as the ranking calculations iterate over the graph several times, until the graph converges, the weighting scores values are recalculated at every iteration and become closer and closer to their actual ranking score.

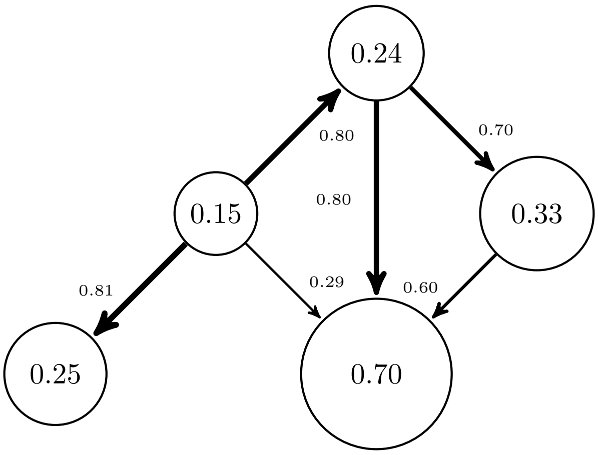


Figure 3.5: A Simplified *TextRank* Graph with Weight Scores

3.1.4 Why the Voting System works in TextRank

The *PageRank* voting model works for keyword extraction and summarization because by linking between co-occurrences a connection is built between similar concepts appearing in the full text. By the voting system, concepts that are re-occurring in the text automatically get voted up since they will receive votes from other elements similar to them selves. The votes are not just votes, they are weighted recommendations based on the importance of the voter. This is why units that are highly connected will vote each other up as their importance will grow as the number of received votes increases, additionally in *TextRank* the strength of the connection between the nodes has an impact on the final ranking, since the stronger the connection is the higher the ranking vote will become.

3.2 TFxIDF

TFxIDF or Term Frequency-Inverted Document Frequency is one of the simpler methods to summarize text. The method is based on the significance of a given word in the full text versus the frequency of the word in the language. The "language" is usually represented by a collection of full text documents. The summary is generated by calculating the *TFxIDF* score of each the terms in each sentence in the text. The sentences with the highest combined *TFxIDF* score are extracted for the summary.

The *TFxIDF* algorithm used in this project is the one listed in (Jurafsky & Martin, 2009).

$$TF_{i,j} = \frac{term_i \in d_j}{terms \in d_j} \tag{3.5}$$

The *TF* in Equation 3.5 stands for term frequency and is the number of times a term i occurs in a document j. For normalization the frequency of a term is divided by the total number of terms in the document.

$$IDF_i = log_2 \frac{D}{df_i} \tag{3.6}$$

The *IDF* in Equation 3.6 stands for inverted term frequency and is introduced for countering the fact that some words have a common high occurrence in any text. The *IDF* will show if a words frequency in a text is significant for the given text or just a common word in the language. The *IDF* of a term i is calculated by dividing the total number of documents D by the number of documents containing the term df_i.

$$weight(w_i) = TF_{i,j} \; * \; IDF_i \tag{3.7}$$

The *TF* and *IDF* are then multiplied together to determine the weight of the given term, as shown in Equation 3.7.

Let us assume, for example, the weight of the word *veðurstofa* in Figure 3.3 is to be calculated. We see that this word only appears once in the text, giving it a TF score of $0.00935 = \frac{1}{107}$, while the word *á* has a TF score of $0.056 = \frac{6}{107}$. However, the word *á* has no significant meaning in the text in the same way *veðurstofa* has. By multiplying the TF score with the IDF score we can reverse this skewness since *á* is a very common word and *veðurstofa* is a more rare word. By calculating the *IDF* for *veðurstofa*, we get $6.3219 = log_2 \frac{80}{1}$, assuming that we found the word *veðurstofa* in 1 out of 80 documents, while the *IDF* for *á* is $0.0181 = log_2 \frac{80}{79}$, assuming we found the word á in 79 out of 80 documents.

The *TFxIDF* for *veðurstofa* then comes out as $0.0591 = 0.00935 * 6.3219$ while it for *á* is $0.0017 = 0.056 * 0.0181$. This way the score of the words in the sentences will be relative to their overall importance in the given text and the whole document collection.

To calculate the weight of a whole sentence all the *TFxIDF* weights for each the terms in the sentence are summed and divided by the length of the sentence as a normalization factor (see Equation 3.8).

$$weight(S_i) = \frac{\sum weight(w_i)}{S_i} \tag{3.8}$$

The *TFxIDF* implementation was implemented in such a way that it could generate the summary from the raw unmodified text or by lemmatizing the words in the text first and use a lemmatized *IDF* word list. When a word is lemmatized it is converted into its base form, also called lemma. For example, the noun *houses* in plural, is lemmatized into *house*.

After a few test runs it became apparent that *TFxIDF* favoured longer sentences. To compensate for this the normalization factor was introduced. This normalization is simply the current sentence weight divided by the current sentence length. This is a simplification of the normalization factor used in (Sharifi, Hutton, & Kalita, 2010). Further improvements were done by restricting the frequency count to only include nouns and adjectives as was done by (Seki, 2002). This improved the quality of the generated summaries tremendously even though the modifications to the original algorithm where minimal.

The *TFxIDF* is a simple and fast to compute method, that gives a good indication of what words in a document have an overall high importance. Even though the *TFxIDF* method in this project has been used as a summarization method on its own, it is often is used as a determination part of other text summarization methods to, for example to identify features for classification for machine learning or for graph-based methods to calculate centroid values (Nenkova & McKeown, 2012).

The *IDF* were generated from 80 articles from the Icelandic newspaper Morgunblaðið. The articles where selected without any requirements of content or length. 80 articles where selected since this would give a 20/80 ratio between the summaries generated and the *IDF* frequency file.

3.3 Baseline

For the baseline a *Lead-Based* summarizer was implemented. This summarizer only selects sentences that appear in the beginning of the text. This is the same as selecting the first n sentences in a text. This approach was chosen, as it has been noted, that this baseline is quite strong and difficult to beat for news articles. The reason is due to the journalistic convention for putting the most important parts of an article in the beginning of the text (Nenkova, 2005).

3.4 Example Summaries

Examples of summaries generated by the implemented methods can be found in Appendix A.

Chapter 4

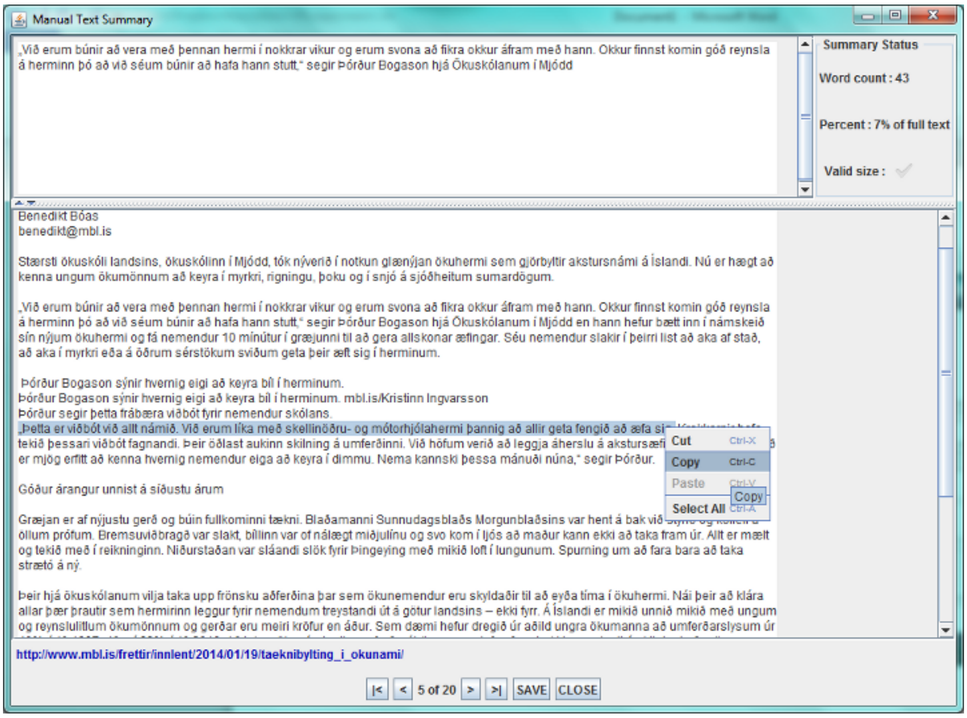
Evaluation

To evaluate the summaries generated by the implemented methods, the summaries had to be compared to summaries created by humans. In general, there is no correct version of a summary but some summaries may be considered to be more accurate and readable than others. The summaries created by humans for comparing to automatically generated summaries are often referred to as *gold standard* summaries or *reference* summaries (Nenkova, 2005, 2006). For comparing the reference summaries and the automatically generated summaries, a standardized tool was needed to make the results comparable to the results presented in the *TextRank* paper and possibly to results found in other papers, if needed. Therefore, the same evaluation tool used in the *TextRank* paper was used to evaluate the summaries generated for this project.

4.1 Reference Summaries

To evaluate the quality of the summaries generated by *TextRank* and *TF-IDF*, reference summaries were created by two different people. The reference summarizers were given 20 articles from mbl.is to summarize. The articles were mostly written in January 2014 and were chosen for their length of around 500-600 words, rather than for their content or authorship. The reference summarizers were instructed to create summaries of at least 100 words, such that the results could be compared more directly to the results presented in the *TextRank* paper, where summaries of 100 words are used for the experiments.

In order to help the reference summarizer to get a better overview of the summarization task, and to help them to create summaries in a more time efficient way, a small GUI program was developed for the task, see Figure 4.1. The design of the manual text summarization program is simple. It has two text areas, labels for word count, summary percentage size and navigation buttons for navigating between the texts to be summarized. The bottom text area shows the full text and in order to create a summary the users(referencers) simply copy text from the full text area into the top text area, which is the summary text area. When the summary has reached a minimum of 100 words and is at least 20% of the original full text, a green check mark becomes visible, letting the user know that the minimum requirements for the summary were fulfilled. The users were instructed that the summaries should be extractive but no instructions where given about the order of the sentences other than this was upto the referencer to decide.



„Við erum búnir að vera með þennan hermi í nokkrar vikur og erum svona að fikra okkur áfram með hann. Okkur finnst komin góð reynsla á herminn þó að við séum búnir að hafa hann stutt," segir Þórður Bogason hjá Ökuskólanum í Mjódd

Word count : 43

Percent : 7% of full text

Valid size : ✓

Benedikt Bóas
benedikt@mbl.is

Stærsti ökuskóli landsins, ökuskólinn í Mjódd, tók nýverið í notkun glænýjan ökuhermi sem gjörbyltir akstursnámi á Íslandi. Nú er hægt að kenna ungum ökumönnum að keyra í myrkri, rigningu, þoku og í snjó á sjóðheitum sumardögum.

„Við erum búnir að vera með þennan hermi í nokkrar vikur og erum svona að fikra okkur áfram með hann. Okkur finnst komin góð reynsla á herminn þó að við séum búnir að hafa hann stutt," segir Þórður Bogason hjá Ökuskólanum í Mjódd en hann hefur bætt inn í námskeið sín nýjum ökuhermi og fá nemendur 10 mínútur í græjunni til að gera allskonar æfingar. Séu nemendur slakir í þeirri list að aka af stað, að aka í myrkri eða á öðrum sérstökum sviðum geta þeir æft sig í herminum.

Þórður Bogason sýnir hvernig eigi að keyra bíl í herminum.
Þórður Bogason sýnir hvernig eigi að keyra bíl í herminum. mbl.is/Kristinn Ingvarsson
Þórður segir þetta frábæra viðbót fyrir nemendur skólans.
„Þetta er viðbót við allt námið. Við erum líka með skellinöðru- og mótorhjólahermi þannig að allir geta fengið að æfa si
tekið þessari viðbót fagnandi. Þeir öðlast aukinn skilning á umferðinni. Við höfum verið að leggja áherslu á akstursæfi
er mjög erfitt að kenna hvernig nemendur eiga að keyra í dimmu. Nema kannski þessa mánuði núna," segir Þórður.

Góður árangur unnist á síðustu árum

Græjan er af nýjustu gerð og búin fullkominni tækni. Blaðamanni Sunnudagsblaðs Morgunblaðsins var hent á bak við
öllum prófum. Bremsuviðbragð var slakt, bíllinn var of nálægt miðjulínu og svo kom í ljós að maður kann ekki að taka fram úr. Allt er mælt og tekið með í reikninginn. Niðurstaðan var sláandi slök fyrir Þingeying með mikið loft í lungunum. Spurning um að fara bara að taka strætó á ný.

Þeir hjá ökuskólanum vilja taka upp frönsku aðferðina þar sem ökunemendur eru skyldaðir til að eyða tíma í ökuhermi. Nái þeir að klára allar þær þrautir sem hermirinn leggur fyrir nemendum treystandi út á götur landsins – ekki fyrr. Á Íslandi er mikið unnið mikið með ungum og reynslulitlum ökumönnum og gerðar eru meiri kröfur en áður. Sem dæmi hefur dregið úr aðild ungra ökumanna að umferðarslysum úr

http://www.mbl.is/frettir/innlent/2014/01/19/taeknibylting_i_okunami/

Figure 4.1: GUI program for manual text summarization.

According to Nenkova (2005) the quality of the human created summaries can depend on the individual human summarizer, but as a whole group, human summarizers outperform system generated summaries. Since only two sets of human generated summaries are used in this experiment it is hard to say if Nenkova's observations apply to this experiment. Radev et. al (2002) observed that humans, generating summaries of even relatively straight forward news articles, only seemed to agreed 60% of the time, when sentence overlap measuring was used. Their observations point out clearly how difficult the task of generating and identifying the ideal summary is.

4.2 Evaluation Toolkit

The evaluation tool *ROUGE* (Recall-Oriented Understudy for Gisting Evaluation) was used to evaluate the summaries generated by *TextRank* and *TF-IDF*. *ROUGE* measures the similarity between the computer generated summaries and the human created summaries by counting the number of overlapping unit n-grams, word sequences and word pairs (Lin & Hovy, 2003; Lin, 2004).

4.2.1 *ROUGE* Settings

Since the texts used are Icelandic it is not possible to use the stemmed and removal of stop words options in the *ROUGE* evaluation tool as was done by Mihalcea & Tarau (2004). However, the basic settings of *ROUGE* give an overview of how summaries generated by *TextRank* and *TF-IDF* compare to the summaries created by humans.

-a	: Evaluate all systems
-n 2	: Compute ROUGE-1, ROUGE-2
-x	: Do not calculate ROUGE-L
-c	: Specify CF% (0 <= CF <= 100) confidence interval to compute.
-r	: Specify the number of sampling point in bootstrap resampling.
-f	: Select scoring formula: 'A' => model average; 'B' => best model.
-p	: Relative importance of recall and precision ROUGE scores.
-t 0	: Use sentence as counting unit.
-d	: Print per evaluation average score for each system.
-l 100	: Only use the first 100 words in the system/peer summary for the evaluation.

```
ROUGE-1.5.5.pl -a -n 2 -x -c 95 -r 1000 -f A -p 0.5 -t 0 -d -l 100 settings.xml
```

Figure 4.2: ROUGE settings.

The results presented by Mihalcea & Tarau (2004) are generated using the n-gram(1, 1) settings which correspond to the ROUGE-1. The ROUGE-1 gives the comparisons results using the uni-gram match of the summaries. When using uni-gram match the generated and the reference summaries are matched a single word at a time to find common words in the summaries. The ROUGE-1 n-gram(1, 1) has been found to give the highest correlation to the human reference summaries with a confidence level of 95% (Lin & Hovy, 2003). Only the first 100 words are considered in the *ROUGE* tests run in (Mihalcea & Tarau, 2004). For summary matching on the Icelandic texts, tests were run comparing both only the first 100 words and the whole generated summary.

The *ROUGE* settings used for the experiments in this project are listed in Figure 4.2. To trunk the summaries down to 100 words the command -l 100 was used.

In order to evaluate the generated summaries with *ROUGE* they were formatted and converted into html files. For this step the Perl script *prepare4rouge* from (Ganesan, 2010) was used. The script converts the summaries to the required html format, sets up the settings file and builds the file structure needed to run the *ROUGE* tests.

ROUGE-1 and ROUGE-2

ROUGE-1 is computed as the count of uni-grams appearing both in system generated summaries and in the reference summaries divided by the number of n-grams in the reference summaries. ROUGE-2 is calculated in the same way, only counting the matches of bi-grams (two consecutive words) instead. This makes ROUGE-N a recall measurement on matching n-grams between system generated summaries and reference summaries (Lin, 2004).

$$ROUGE-N = \frac{\sum\limits_{S\in\{Reference\,Summaries\}} \sum\limits_{gram_n\in S} Count_{match}(gram_n)}{\sum\limits_{S\in\{Reference\,Summaries\}} \sum\limits_{gram_n\in S} Count(gram_n)} \tag{4.1}$$

In Figure 4.1 $gram_n$ stands for the length of the n-gram, indicating if it is a uni-gram, bi-gram, etc., and $Count_{match}(gram_n)$ is the number of n-grams co-occurring both in a system generated summary and in the set of reference summaries.

Since ROUGE co-occurrence measurement is based on content overlap, it is possible to determine if the same general concepts are discussed in an automatic generated summary and in a manually created reference summary. The *ROUGE* n-gram matching cannot, however, determine if the matched results are coherent or if the sentence flow is even sensible.

Recall, Precision and F-measure

The ROUGE-N scores come out as three measurements, namely recall, precision and F-measure. Recall is the fraction of n-grams in the reference summaries that are found in the automatically generated summary (see Formula 4.2 and Figure 4.3), this returns the fraction of selected items that are relevant.

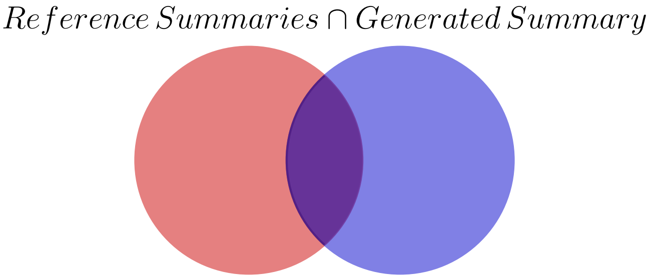


Figure 4.3: The matched n-grams are the Intersection of Reference Summaries and the Generated Summary.

$$Recall = \frac{|matches \in \{Reference\,Summary\} \cap matches \in \{Generated\,Summary\}|}{|matches \in \{Reference\,Summary\}|} \tag{4.2}$$

Precision, on the other hand, is the fraction of n-grams in the generated summary that are present in the reference summary, this gives the percentage of relevant items that are selected. See the Formula 4.3 and Figure 4.3.

$$Precision = \frac{|matches \in \{Reference\,Summary\} \cap matches \in \{Generated\,Summary\}|}{|matches \in \{Generated\,Summary\}|} \tag{4.3}$$

Ideally, there should be a balance between the two measurement, such that no false matches are included and no positive matches are excluded but there is always a trade-off between the two. The F-measure is an average measurement between the recall and precision, giving a more general measurement to relate to.

All the scores from the comparisons evaluations are given as recall, precision and F-measure, see Table 4.1 and 4.2 for results.

4.3 Results

The results from the *ROUGE* co-occurrence matching between the reference summaries and the system generated summaries are listed in Table 4.1 and 4.2. Evaluation was done on both truncated and whole length summaries.

The results in 4.1 are as such identical to the results presented by Mihalcea & Tarau (2004). The results presented in the *TextRank* paper show a ROUGE-1 score of 47.08%. This score is consistent with the ROUGE-1 F-measure score of 47.48% measured on the implemented *TextRank* method for Icelandic texts. The tiny difference can result due to the fact that we have a different set of full texts or the low number of reference summaries. The authors of the *TextRank* paper do not publish their ROUGE-2 scores but in comparing our *TextRank* ROUGE-2 scores towards the baseline ROUGE-2 scores, it is obvious that here the baseline is even harder to beat. However, it is interesting to note that the ROUGE-2 scores in Table 4.1 and 4.2 show that the recall stays pretty much the same when the evaluation is run on truncated and non-truncated summaries.

The results show that baseline is difficult to beat since none of the implemented methods manage to beat it. This result is although not totally surprising and was discussed in Section 3.3, where it was pointed out that for newspaper articles a baseline of selecting the first few sentences in the article was very strong due to the journalistic convention of including the most informative parts in the first few sentences. Figure 4.4 and 4.5 do, though, show that it is possible to beat the baseline on individual texts from the article collection.

Looking at the scores in the two tables 4.1 and 4.2, it is noticeable that the scores for the baseline and the other methods differ on recall and precision. The baseline is higher on recall, while the other methods are higher on precision. The higher recall shows that for the baseline there are selected a higher number of n-grams marked as matches in the generated summaries resulting in a higher coverage. While the higher percentage for precision for the other methods shows that the correct matching of n-grams found is higher. The relationship between these two measurements is always a balance but they give a good indication of the quality of the matching performed towards the reference summaries.

Looking at the results in the two tables we see that the score results in Table 4.2 are all generally higher. This corresponds with the results and observations done in (Delort & Alfonseca, 2011) where it is noted that there is a relationship between the length of the summaries generated and the ROUGE score.

4.3.1 Comparing the Methods

The test results show that the *TextRank* method performs better than the *TF-IDF* method but overall the difference between them is not overwhelmingly great. It is hard to speculate why this is the case but in general these are two simple methods, that both use a similar syntactic filter. The *TextRank* method uses the filter to identify similarities between sentences and the *TF-IDF* uses the filter to find words to calculate the *TF-IDF* weight scores on. This can explain the similarity in the results, since the number of words

	ROUGE-1: uni-gram			ROUGE-2: bi-gram		
	Recall	Precision	F-measure	Recall	Precision	F-measure
Baseline	0.57757	0.51711	0.54532	0.44203	0.39523	0.41706
TextRank	0.45076	0.50247	0.47482	0.27223	0.30299	0.28651
TFxIDF	0.44531	0.46435	0.45424	0.26563	0.27754	0.27119
TFxIDF lemmatized	0.42125	0.42452	0.42250	0.22631	0.22766	0.22678

Table 4.1: Table of the results of ROUGE-1 (n-gram(1, 1)) and ROUGE-2 matching first 100 words only.

	ROUGE-1: uni-gram			ROUGE-2: bi-gram		
	Recall	Precision	F-measure	Recall	Precision	F-measure
Baseline	0.56658	0.52570	0.54376	0.41302	0.38233	0.39589
TextRank	0.47585	0.55199	0.50932	0.29931	0.34852	0.32088
TFxIDF	0.45986	0.50116	0.47759	0.27808	0.30442	0.28925
TFxIDF lemmatized	0.45221	0.47283	0.46158	0.25574	0.26905	0.26187

Table 4.2: Table of the results of ROUGE-1 (n-gram(1, 1)) and ROUGE-2 matching on whole summaries.

matching the filter are limited and both methods would end up using the same words for calculating the weighting scores. This observation is not enough to draw any final conclusions from but could be the basis of more experiments to find improvements in the implementations.

The surprise is that the lemmatized version of *TF-IDF* does not perform better or the same as the "regular" version of *TF-IDF*. It could be expected, that the lemmatized version had a better performance, since there would be a higher number of matching terms for every term in the lemmatized version.

4.3.2 Variations for the Different Texts

In Figure 4.4 and 4.5, the F-measure for ROUGE-1 and ROUGE-2 is plotted for each of the twenty generated summaries. The bar plots give an overview of the difference in scores the methods give for the individual news articles. The Figures show that there is an overlap of the scores for each of the methods for the whole range of articles but also show that the baseline is quite dominating. The score variation for the articles varies quite a bit, for example for some articles there is an score measurement difference of around 30% between the highest and lowest score. For ROUGE-2 this score difference is even bigger, being over 50% for some articles. This result is expected as the ROUGE-2 score requires more accurate overlay matching between the automatically generated summaries and the reference summaries. Furthermore, the results in the two plots show the importance in having an appropriate number of reference summaries to get a realistic average score.

4.3.3 Discussion of Results

The overall results are satisfying, even though is was not possible to beat the baseline. The evaluation results for the *TextRank* algorithm implemented for Icelandic texts gave results that where identical to the ones presented in the *TextRank* paper. This tells us that it is possible to implement a generic unsupervised text summarizer for Icelandic without loss of quality. The Icelandic IceNPL POS-tagger was successfully used for the syntactic filter in both *TextRank* and *TFxIDF*. For the *TFxIDF* method the addition of the filter, improved the result tremendously, showing that the idea of using a syntactic filter is valid when generating summaries.

The experiments for this project were only performed on twenty news articles chosen from the news site mbl.is and each of the twenty texts were paired with two reference summaries created by humans. This is a small test set but looking at the results the tendencies are apparent and a clear conclusion can be made of which summarizing method performed best. The results reconfirm that for this type of texts the Lead-based baseline is hard to beat.

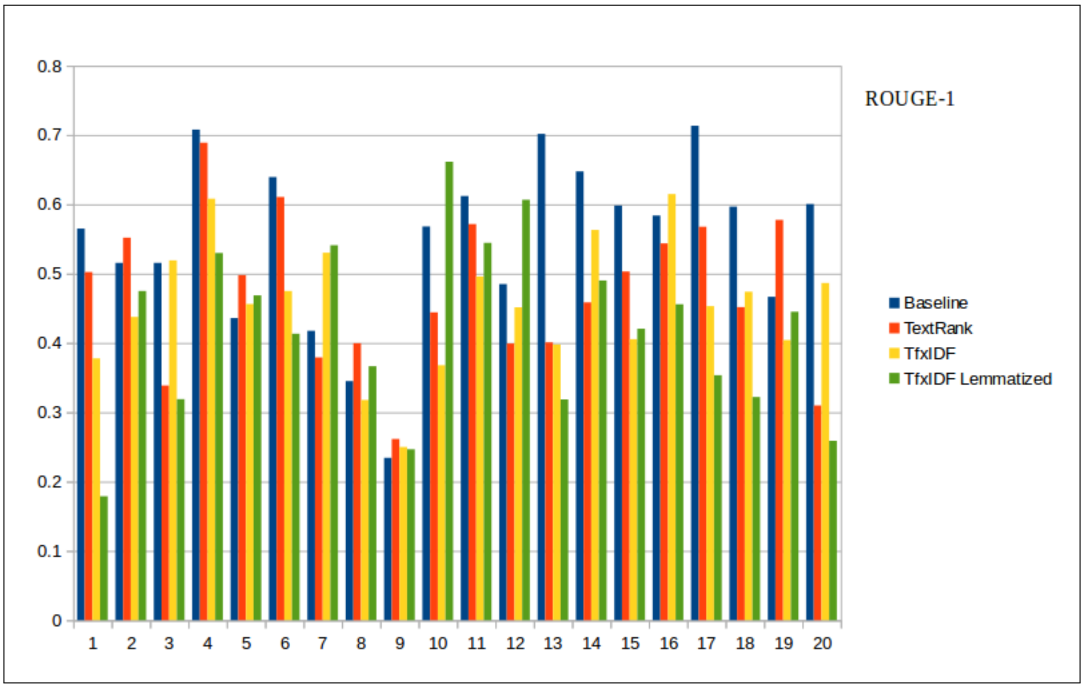


Figure 4.4: ROUGE-1 Plot - summaries truncated to 100 words

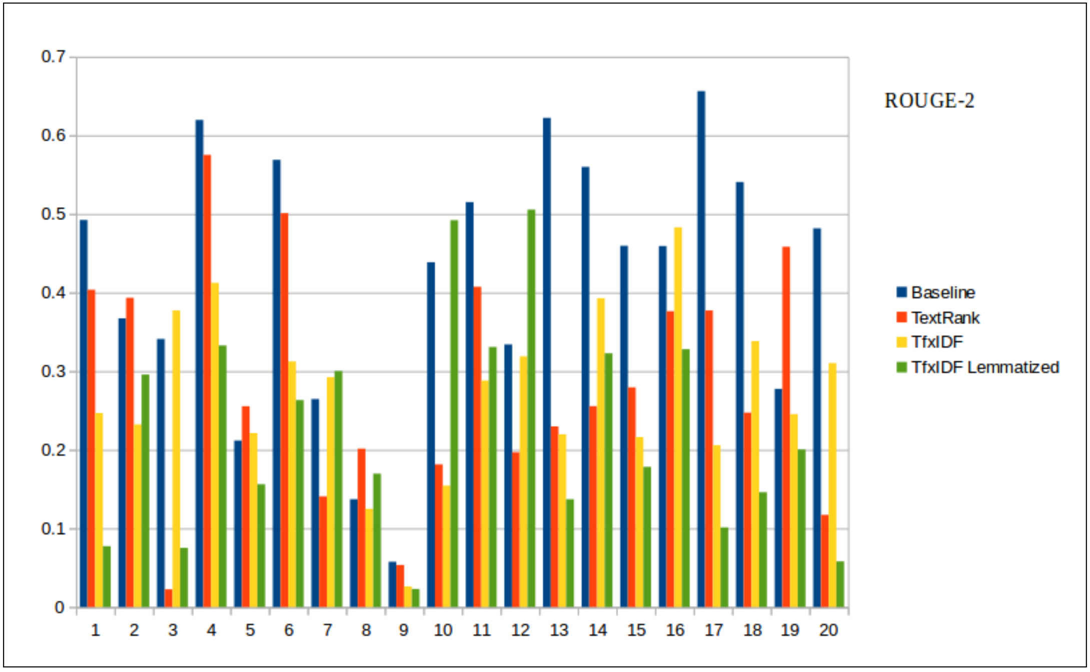


Figure 4.5: ROUGE-2 Plot - summaries truncated to 100 words

Chapter 5

Future Work

The overall results where satisfying, even though is was not possible to beat the baseline. The goal was to generate generic readable summaries from Icelandic texts and this has been achieved. There is room for improvement in the implemented methods and it is clear that other and more extensive methods than the two implemented for this project are needed. In (Dalianis, 2000) it was suggested that always including the heading in the summary when summarizing news articles improves the summary. This approach would mean that the implemented summarizers would no longer be generic but would be bound to the news article genre. An other interesting approach mentioned in Chapter 2 was the use of Wikipedia for topic identification in the text, this would definitely be an interesting technique to experiment with. Also, it would be interesting to add the IceWordNet to the summarizers to see if this would improve the results. Currently, the IceWordNet project is discontinued but it would be interesting to see if it would be useful for text summarization in its current state.

One of the future goals would also be to collect more human created reference summaries, as only having two reference summaries for each of the twenty summaries used in the evaluation in this project is not enough. The use of twenty full texts for the summarization task is also a very low number, compared to the data sets used at the DUC 2002 conference where 567 full texts with pairing reference summaries where used. Creating a larger set of full text and summary pairs would also benefit future effort towards experimenting with a supervised text summarization methods.

Currently the implemented summarizer can only read in clean formatted text for summarization, extending it to stripping html tags from texts would give the users the possibility to feed the summarizer with the url of the text they wanted summarized, would increase the usability tremendously.

Chapter 6

Conclusions

To the goal of this project was to examine the area of text summarization and select one method or more methods for a prototype implementation. The prototype should be able to automatically generate readable summaries of Icelandic texts. In addition, the summaries generated should be evaluated in a standardized way to assess the quality of them. A brief outline of the main outcomes of this project are:

- An overview was given of the area of text summarization. Some types of summarizes and approaches towards text summarization where looked into. This gave an angle of approaches possible for implementing a text summarizer tailored for the Icelandic language.
- A summarizer prototype was implemented cable of automatically generating summaries of Icelandic texts. The summarizer can generate extractive generic summaries using the unsupervised methods *TextRank* and *TFxIDF*.
- The generated summaries where compared to human created summaries and an quality measurement comparison was done between the two sets of summaries. The results for the *TextRank* method matched the results presented in the *TextRank* paper. The score for the *TFxIDF* method were a bit lower than for the *TextRank* method. However, the baseline summarizer had the highest score of all the implemented methods, showing that the *Lead-based* baseline is hard to beat for news articles.

The goal of this project to implement a prototype summarizer has been achieved but further improvements of the code, discussed in Chapter 5 would improve the usability of the summarizer. Additionally, some of the more advanced methods, some of which depend on external lexical sources, do present an interesting approach towards how the quality of the implemented summarizer could be improved.

Appendices

Appendix A

A.1 Example Summaries

Method	Summary
TextRank	Bændur um allt Norðurland eru á tánum vegna illviðrisspár um komandi helgi. Þórarinn segir að þegar sé byrjað að tala við þann mannskap sem farið hefur í göngur síðustu haust þannig að menn verði í startholunum ef þurfa þykir í fyrramálið. "Sumir eru ekki búnir að heyja, ætluðu sér kannski að ná seinni slætti, þannig að aðstæður manna eru misjafnar en menn komast nú alveg í gegnum það held ég." Göngur gætu hafist strax á morgun. "Við tökum enga sénsa á svona löguðu. Víða er fundað í kvöld og svo gæti farið að göngur hefjist sumstaðar strax á morgun.
TFxIDF	Veðurstofa Íslands spáir norðanhríð á föstudag með slyddu eða snjókomu í 150-250 metra hæð yfir sjávarmáli og vindhraða allt að 15-23 m/s. Hann segir bændur í Höfðahverfi ætla að taka stöðuna eftir veðurspár kvöldsins, enn geti brugðið til beggja vona. Birgir er formaður Félags Sauðfjárbænda við Eyjafjörð og ætlar að heyra í mönnum í sveitinni í kvöld. "Sumir eru ekki búnir að heyja, ætluðu sér kannski að ná seinni slætti, þannig að aðstæður manna eru misjafnar en menn komast nú alveg í gegnum það held ég." Aðspurður segir Þórarinn allan gang á því hversu vel menn séu undir það búnir að taka féð heim á tún svo snemma.

TFxIDF, lemmatized	Veðurstofa Íslands spáir norðanhríð á föstudag með slyddu eða snjókomu í 150-250 metra hæð yfir sjávarmáli og vindhraða allt að 15-23 m/s. Á laugardagsmorgun er svo von á norðvestan 18-25 m/s á Norður- og Austurlandi og mikilli rigningu neðan við 100-200 metrum yfir sjávarmáli, en annars slyddu eða snjókomu Fjallskilastjórar funda víða í kvöld. Við tökum enga sénsa á svona löguðu. Þannig að manni finnst algjörlega fáránlegt í rauninni að hugsa til þess að svona veður sé framundan, og alveg ótímabært." Ekki stóð til að smala í Eyjafirðinum fyrr en helgina 7. - 8. september og á Vaðlaheiði ekki fyrr en 14. september
Baseline	Bændur á tánum vegna illviðrisspár. http://www.mbl.is / frettir / innlent / 2013/08/26/baendur_ a_ tanum_ vegna_ illvidrisspar/. Una Sighvatsdóttir. una@mbl.is. mbl.is. "Ég held að menn geti ekki leyft sér að sitja bara heima þegar það spáir svona, "segir Birgir H. Arason sauðfjárbóndi á Gullbrekku í Eyjafirði. Bændur um allt Norðurland eru á tánum vegna illviðrisspár um komandi helgi. Víða er fundað í kvöld og svo gæti farið að göngur hefjist sumstaðar strax á morgun. Veðurstofa Íslands spáir norðanhríð á föstudag með slyddu eða snjókomu í 150-250 metra hæð yfir sjávarmáli og vindhraða allt að 15-23 m/s. Á laugardagsmorgun er svo von á norðvestan 18-25 m/s á Norður- og Austurlandi og mikilli rigningu neðan við 100-200 metrum yfir sjávarmáli, en annars slyddu eða snjókomu.

Table A.1: Example Summaries Generated[1]

[1] Note that these summaries have been generated from text files encoded with UTF-8 with BOM. Using other encodings may result in different result for summaries generated with TextRank.

	Summary
Reference Summary 1	Bændur á tánum vegna illviðrisspár. Bændur um allt Norðurland eru á tánum vegna illviðrisspár um komandi helgi. Víða er fundað í kvöld og svo gæti farið að göngur hefjist sumstaðar strax á morgun. Óveðrið í september fyrir ári síðan er mönnum að sjálfsögðu í fersku minni en þá drápust yfir 3.500 kindur í Þingeyjarsýslu og Eyjafirði. „Við tökum enga sénsa á svona löguðu. Menn eru með svona frekar neikvæðan fiðring núna, horfandi á allar veðurspár og hringjandi hver í annan fram og til baka," segir Þórarinn Ingi Pétursson, bóndi á Grýtubakka í Höfðahverfi. Þórarinn segir að þegar sé byrjað að tala við þann mannskap sem farið hefur í göngur síðustu haust þannig að menn verði í startholunum ef þurfa þykir í fyrramálið.
Reference Summary 2	Bændur um allt Norðurland eru á tánum vegna illviðrisspár um komandi helgi. Víða er fundað í kvöld og svo gæti farið að göngur hefjist sumstaðar strax á morgun. Veðurstofa Íslands spáir norðanhríð á föstudag með slyddu eða snjókomu í 150-250 metra hæð yfir sjávarmáli og vindhraða allt að 15-23 m/s. Á laugardagsmorgun er svo von á norðvestan 18-25 m/s á Norður- og Austurlandi og mikilli rigningu neðan við 100-200 metrum yfir sjávarmáli, en annars slyddu eða snjókomu. Óveðrið í september fyrir ári síðan er mönnum að sjálfsögðu í fersku minni en þá drápust yfir 3.500 kindur í Þingeyjarsýslu og Eyjafirði.

Table A.2: Manually Created Reference Summaries (used in evaluation)

A.2 Full Text

Bændur á tánum vegna illviðrisspár.
http://www.mbl.is/frettir/innlent/2013/08/26/baendur_a_tanum_vegna_illvidrisspar/.
Una Sighvatsdóttir.
una@mbl.is.
mbl.is.

"Ég held að menn geti ekki leyft sér að sitja bara heima þegar það spáir svona," segir Birgir H. Arason sauðfjárbóndi á Gullbrekku í Eyjafirði. Bændur um allt Norðurland eru á tánum vegna illviðrisspár um komandi helgi. Víða er fundað í kvöld og svo gæti farið að göngur hefjist sumstaðar strax á morgun.

Veðurstofa Íslands spáir norðanhríð á föstudag með slyddu eða snjókomu í 150-250 metra hæð yfir sjávarmáli og vindhraða allt að 15-23 m/s. Á laugardagsmorgun er svo von á norðvestan 18-25 m/s á Norður- og Austurlandi og mikilli rigningu neðan við 100-200 metrum yfir sjávarmáli, en annars slyddu eða snjókomu.

Óveðrið í september fyrir ári síðan er mönnum að sjálfsögðu í fersku minni en þá drápust yfir 3.500 kindur í Þingeyjarsýslu og Eyjafirði.

Fjallskilastjórar funda víða í kvöld

"Manni líst hreinlega ekkert á þetta," segir Birgir, sem var í miðjum slætti þegar blaðamaður mbl.is heyrði í honum nú laust fyrir kvöldmat. "Við erum að reyna að klára heyskapinn, á síðustu metrum í seinni slætti og sumir eru enn ekki byrjaðir. Þannig að manni finnst algjörlega fáránlegt í rauninni að hugsa til þess að svona veður sé framundan, og alveg ótímabært."

Ekki stóð til að smala í Eyjafirðinum fyrr en helgina 7. - 8. september og á Vaðlaheiði ekki fyrr en 14. september. Veðurspáin setur áætlanir bænda því mjög úr skorðum. "Það bjóst náttúrulega enginn við svona ósköpum," segir Birgir en bætir við að jákvæða hliðin sé að menn eru nú reynslunni ríkari síðan í fyrra.

Birgir er formaður Félags Sauðfjárbænda við Eyjafjörð og ætlar að heyra í mönnum í sveitinni í kvöld. Og það er víðar sem fundað verður vegna veðurspárinnar, því samkvæmt heimildum mbl.is hafa fjallskilastjórar m.a. verið boðaðir á fund í Skagafirði og Þingeyjarsýslu.

Göngur gætu hafist strax á morgun

"Við tökum enga sénsa á svona löguðu. Menn eru með svona frekar neikvæðan fiðring núna, horfandi á allar veðurspár og hringjandi hver í annan fram og til baka," segir Þórarinn Ingi Pétursson, bóndi á Grýtubakka í Höfðahverfi og formaður Landssamtaka sauðfjárbænda.

Hann segir bændur í Höfðahverfi ætla að taka stöðuna eftir veðurspár kvöldsins, enn geti brugðið til beggja vona. "Það er náttúrulega bara þessi eina spá sem kominn er, en spár fara að skýrast betur í kvöld. Ef það stefnir í þetta veður þá reikna ég nú frekar með því að við rjúkum af stað á morgun, margir hverjir."

Þórarinn segir að þegar sé byrjað að tala við þann mannskap sem farið hefur í göngur síðustu haust þannig að menn verði í startholunum ef þurfa þykir í fyrramálið.

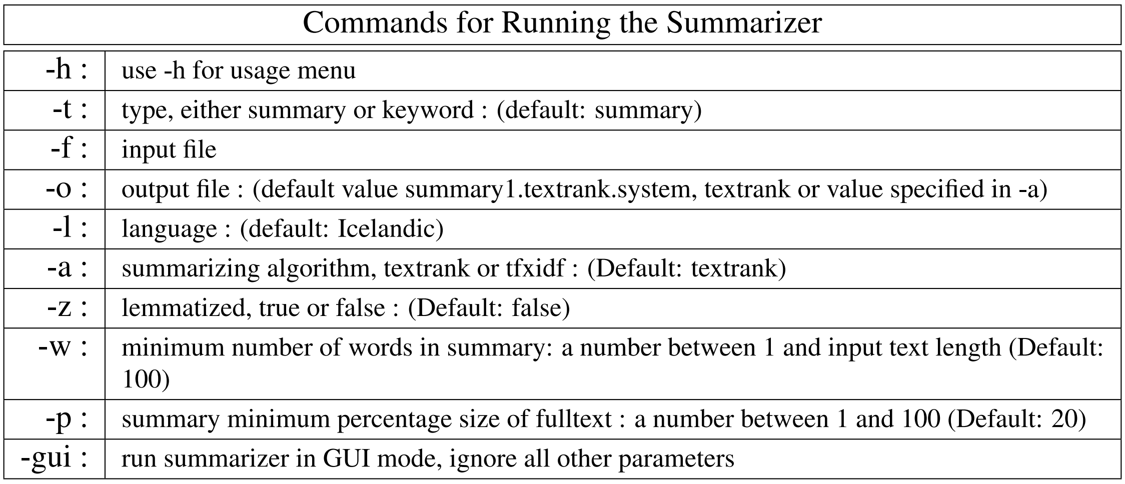
Aðspurður segir Þórarinn allan gang á því hversu vel menn séu undir það búnir að taka féð heim á tún svo snemma. "Sumir eru ekki búnir að heyja, ætluðu sér kannski að ná seinni slætti, þannig að aðstæður manna eru misjafnar en menn komast nú alveg í gegnum það held ég."

Table A.3: The Full Text the Summaries are Generated from

Appendix B

B.1 How to Run the Summarizer

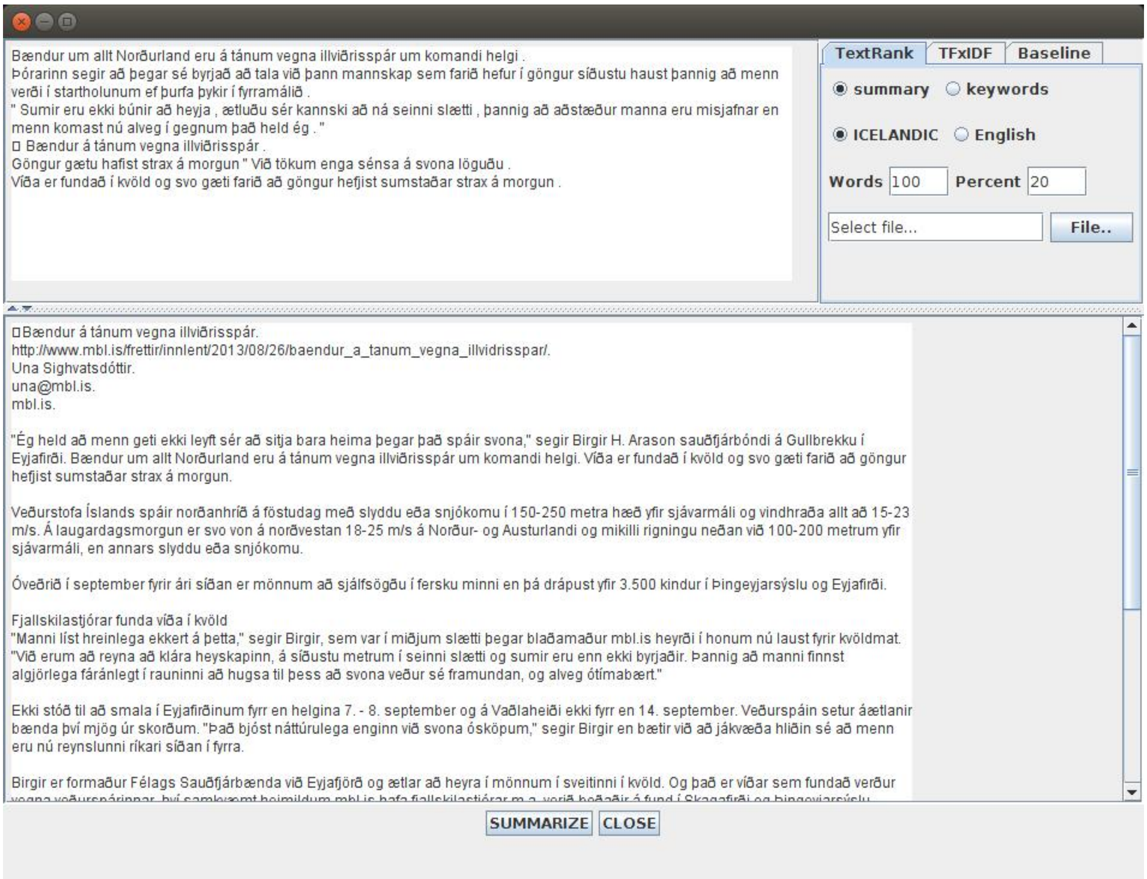
B.1.1 Running the Summarizer from the Command Line



Commands for Running the Summarizer	
-h :	use -h for usage menu
-t :	type, either summary or keyword : (default: summary)
-f :	input file
-o :	output file : (default value summary1.textrank.system, textrank or value specified in -a)
-l :	language : (default: Icelandic)
-a :	summarizing algorithm, textrank or tfxidf : (Default: textrank)
-z :	lemmatized, true or false : (Default: false)
-w :	minimum number of words in summary: a number between 1 and input text length (Default: 100)
-p :	summary minimum percentage size of fulltext : a number between 1 and 100 (Default: 20)
-gui :	run summarizer in GUI mode, ignore all other parameters

Figure B.1: Running the Summarizer from Command Line.

B.1.2 The Summarizer Graphical User Interface



Bændur um allt Norðurland eru á tánum vegna illviðrisspár um komandi helgi .
Þórarinn segir að þegar sé byrjað að tala við þann mannskap sem farið hefur í göngur síðustu haust þannig að menn verði í startholunum ef þurfa þykir í fyrramálið .
" Sumir eru ekki búnir að heyja , ætluðu sér kannski að ná seinni slætti , þannig að aðstæður manna eru misjafnar en menn komast nú alveg í gegnum það held ég . "
□ Bændur á tánum vegna illviðrisspár .
Göngur gætu hafist strax á morgun " Við tökum enga sénsa á svona löguðu .
Víða er fundað í kvöld og svo gæti farið að göngur hefjist sumstaðar strax á morgun .

● summary ○ keywords

● ICELANDIC ○ English

Words 100 Percent 20

Select file...

□Bændur á tánum vegna illviðrisspár.
http://www.mbl.is/frettir/innlent/2013/08/26/baendur_a_tanum_vegna_illvidrisspar/.
Una Sighvatsdóttir.
una@mbl.is.
mbl.is.

"Ég held að menn geti ekki leyft sér að sitja bara heima þegar það spáir svona," segir Birgir H. Arason sauðfjárbóndi á Gullbrekku í Eyjafirði. Bændur um allt Norðurland eru á tánum vegna illviðrisspár um komandi helgi. Víða er fundað í kvöld og svo gæti farið að göngur hefjist sumstaðar strax á morgun.

Veðurstofa Íslands spáir norðanhríð á föstudag með slyddu eða snjókomu í 150-250 metra hæð yfir sjávarmáli og vindhraða allt að 15-23 m/s. Á laugardagsmorgun er svo von á norðvestan 18-25 m/s á Norður- og Austurlandi og mikilli rigningu neðan við 100-200 metrum yfir sjávarmáli, en annars slyddu eða snjókomu.

Óveðrið í september fyrir ári síðan er mönnum að sjálfsögðu í fersku minni en þá drápust yfir 3.500 kindur í Þingeyjarsýslu og Eyjafirði.

Fjallskilastjórar funda víða í kvöld
"Manni líst hreinlega ekkert á þetta," segir Birgir, sem var í miðjum slætti þegar blaðamaður mbl.is heyrði í honum nú laust fyrir kvöldmat. "Við erum að reyna að klára heyskapinn, á síðustu metrum í seinni slætti og sumir eru enn ekki byrjaðir. Þannig að manni finnst algjörlega fáránlegt í rauninni að hugsa til þess að svona veður sé framundan, og alveg ótímabært."

Ekki stóð til að smala í Eyjafirðinum fyrr en helgina 7. - 8. september og á Vaðlaheiði ekki fyrr en 14. september. Veðurspáin setur áætlanir bænda því mjög úr skorðum. "Það bjóst náttúrulega enginn við svona ósköpum," segir Birgir en bætir við að jákvæða hliðin sé að menn eru nú reynslunni ríkari síðan í fyrra.

Birgir er formaður Félags Sauðfjárbænda við Eyjafjörð og ætlar að heyra í mönnum í sveitinni í kvöld. Og það er víðar sem fundað verður vegna veðurspárinnar, því samkvæmt heimildum mbl.is hafa fjallskilastjórar m.a. verið boðaðir á fund í Skagafirði og Þingeyjarsýslu

Figure B.2: The Graphical Interface for the Text Summariser.

Bibliography

Barzilay, R., & Elhadad, M. (1997). Using lexical chains for text summarization. In *Proceedings of the ACL workshop on intelligent scalable text summarization.*

Barzilay, R., Elhadad, N., & McKeown, K. R. (2002). Inferring strategies for sentence ordering in multidocument news summarization. *J. Artif. Int. Res.*, *17*(1), 35–55.

Dalianis, H. (2000). *SweSum - a text summarizer for swedish* (Tech. Rep.). KTH.

Das, D., & Martins, A. F. T. (2007). *A survey on automatic text summarization* (Tech. Rep.). Literature Survey for the Language and Statistics II course at Carnegie Mellon University.

Delort, J.-Y., & Alfonseca, E. (2011). Description of the google update summarizer at TAC-2011. In *Proceedings of the text analysis conference 2011 (TAC 2011).*

Edmundson, H. P. (1969). New methods in automatic extracting. *J. ACM*, *16*(2), 264–285.

Ganesan, K. (2010). *prepare4rouge - Script to Prepare for Rouge Evaluation.* Retrieved from `http://kavita-ganesan.com/content/prepare4rouge-script-prepare-rouge-evaluation` (accessed: 05/01-2014)

Hovy, E., & Lin, C.-Y. (1998). Automated text summarization and the SUMMARIST system. In *Proceedings of a workshop on held at baltimore, maryland: October 13-15, 1998.* Stroudsburg, PA, USA.

Jurafsky, D., & Martin, J. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, NJ, USA: Pearson Prentice Hall/Pearson education international.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, *46*(5), 604–632.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the acl-04 workshop.* Barcelona, Spain.

Lin, C.-Y., & Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology - volume 1.* Stroudsburg, PA, USA.

Litvak, M., & Last, M. (2008). Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on multi-source multilingual information extraction and summarization.* Stroudsburg, PA, USA.

Loftsson, H., & Rögnvaldsson, E. (2007). IceNLP: A Natural Language Processing Toolkit for Icelandic. In *Proceedings of interspeech 2007, special session: "speech and language technology for less-resourced languages".* Antwerp, Belgium.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM J. Res. Dev.*, *2*(2), 159–165.

Marcu, D. (1999). The automatic construction of large-scale corpora for summarization research. In *Proceedings of the $22nd$ annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA.

McKeown, K. R., Klavans, J. L., Hatzivassiloglou, V., Barzilay, R., & Eskin, E. (1999). Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of the sixteenth national conference on artificial intelligence and the eleventh innovative applications of artificial intelligence conference innovative applications of artificial intelligence.* Menlo Park, CA, USA.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into texts. In *Proceedings of EMNLP 2004.* Barcelona, Spain.

Nastase, V. (2008). Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *Proceedings of the conference on empirical methods in natural language processing.* Stroudsburg, PA, USA.

Nastase, V., Milne, D., & Filippova, K. (2009). Summarizing with encyclopedic knowledge. In *Proceedings of the second text analysis conference (TAC 2009).*

Nenkova, A. (2005). Automatic text summarization of newswire: Lessons learned from the document understanding conference. In *Proceedings of the $20th$ national conference on artificial intelligence (AAAI 2005) - volume 3.*

Nenkova, A. (2006). Summarization evaluation for text and speech: issues and approaches. In *INTERSPEECH 2006.*

Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In C. C. Aggarwal & C. Zhai (Eds.), *Mining Text Data.* Springer.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The pagerank citation ranking: Bringing order to the web.* (Technical Report No. 1999-66). Stanford InfoLab.

Radev, D., Allison, T., Blair-Goldensohn, S., Blitzer, J., Çelebi, A., Dimitrov, S., ...

Zhang, Z. (2004). MEAD — A platform for multidocument multilingual text summarization. In *Conference on language resources and evaluation (LREC).* Lisbon, Portugal.

Radev, D. R., Hovy, E., & McKeown, K. (2002). Introduction to the special issue on summarization. *Comput. Linguist.*, *28*(4), 399–408.

Saggion, H., & Lapalme, G. (2002). Generating indicative-informative summaries with sumUM. *Comput. Linguist.*, *28*(4), 497–526.

Seki, Y. (2002). Sentence extraction by tf/idf and position weighting from newspaper articles. In *Proceedings of the 3rd national institute of informatics test collection information retrieval (NTCIR) workshop.*

Sharifi, B., Hutton, M.-A., & Kalita, J. K. (2010). Experiments in microblog summarization. In *Proceedings of the 2010 IEEE second international conference on social computing.* Washington, DC, USA.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology.* Edmonton, Canada.

School of Computer Science
Reykjavík University
Menntavegi 1
101 Reykjavík, Iceland
Tel. +354 599 6200
Fax +354 599 6201
www.reykjavikuniversity.is
ISSN 1670-8539