Automatic text summarization of Swedish news articles

Niko Lehto
Mikael Sjödin

Tutor, Anders Fröberg
Examinator, Erik Berglund
Automatic text summarization of Swedish news articles

Niko Lehto  
Linköping University  
Linköping, Sweden  
nikle369@gmail.com

Mikael Sjödin  
Linköping University  
Linköping, Sweden  
mikael.l.sjodin@gmail.com

ABSTRACT
With an increasing amount of textual information available there is also an increased need to make this information more accessible. Our paper describes a modified TextRank model and investigates the different methods available to use automatic text summarization as a means for summary creation of Swedish news articles.

To evaluate our model we focused on intrinsic evaluation methods, in part through content evaluation in the form of measuring referential clarity and non-redundancy, and in part by text quality evaluation measures, in the form of keyword retention and ROUGE evaluation.

The results acquired indicate that stemming and improved stop word capabilities can have a positive effect on the ROUGE scores. The addition of redundancy checks also seems to have a positive effect on avoiding repetition of information. Keyword retention decreased somewhat, however. Lastly all methods had some trouble with dangling anaphora, showing a need for further work within anaphora resolution.

Author Keywords
Automatic text summarization; Language technology; Summary evaluation; Natural language processing.

INTRODUCTION
The explosion of information happening in the recent years has increased textual information available drastically. To make use of all this information we need to find ways to access this information in more efficient manner. One way to achieve part of this is via the usage of automatic text summaries to condense the information. [1] This could be used e.g. to compress information on news sites and articles for easier browsing, to create summaries that is more easily viewable on mobile devices [1] or to generate snippets of the content of web pages for search engine results [2].

A summary can, according to Radev et al. [3], be defined as “a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that.”.

Automatic text summarization is the task to summarize a text with software while preserving the overall meaning and key points. There are two general approaches to automatic text summarization; extraction based methods and abstraction based methods. [4] Extractive summarization extracts sentences without modification based on calculating the importance score of each sentence [5]. This importance score is based on how well this sentence explains the important topics of the text [4]. Abstractive summarization works by generating new sentences based on the source material thus having the potential to generate more high quality summaries [1]. In comparison extractive methods is much simpler to implement than abstractive methods. Extractive methods are explained further under theory and related work. We chose to modify an implementation of an extractive method in our work and thus the focus of this paper will be on extractive based methods.

When it comes to evaluation there are several ways to evaluate if a summary is good, separated into two main categories. Firstly we have extrinsic evaluation which evaluates how helpful this summary is to a specific task. The second is intrinsic evaluation which is an analysis of the actual summary. [6] In our study we focus on the intrinsic evaluation of our summaries.

Intrinsic evaluation can further be divided into content evaluation and text quality evaluation. Content evaluation looks at the presence of important keywords and topics of the summary whereas text quality evaluation looks at the readability of a summary. [6], [7] Some of the linguistic quality aspects text quality evaluation focuses on are: non-redundancy, referential clarity and grammaticality. [7]

To evaluate the text quality of the summaries we will look at referential clarity and redundancy. Referential clarity can be measured by manually counting the number of dangling anaphora in our summaries. An anaphora is a word that refers to an entity earlier in the text, e.g. “He” and “Them”. A dangling anaphora is an anaphora that does not have it’s referred entity included in the summary. [8] The redundancy which is text that repeats information will be measured by counting sentences that are too similar on a lexical level [9].

For the content evaluation we will use ROUGE, Recall-Oriented Understudy for Gisting Evaluation, and an iMatrics keyword extractor. ROUGE is an automated evaluation model which compares the content overlap
between a summary and an human made “ideal” summary. [10]

As for keyword retention, both the source article and the summary will be run through a keyword extractor iMatrics has developed, the overlap of the extracted keywords from the source text and the summary will then be examined.

Aim/Purpose
iMatrics sees the need for automatic summarization and wants to make the information available on different news sites more accessible with the use of automatic text summarization. This is why we will implement and modify a variant of a TextRank summarizer, hereafter referred to as Modified TextRank.

Research questions
Q1: To what degree does our summarized articles preserve the keywords from the iMatrics keyword extractor?
Q2: How does the textual quality of the generated summaries differ, with regards to dangling anaphora and non-redundancy, between the Modified TextRank and the summarizers used for comparison?
Q3: How does Modified TextRank perform with regards to the ROUGE metric compared to other summarizers?

Delimitations
We chose to focus on extractive methods for text summarization because it is a field with extensive research and is much less time consuming to implement compared to the abstractive method.

THEORY AND RELATED WORK
This section will firstly discuss the steps of extractive text summarization in general and then discuss each method we have used in more detail afterwards. Lastly we will discuss how we evaluated these methods.

Extractive summarization
As mentioned earlier extractive summarization works by extracting the most important sentences from a text to make a summary. The basic steps extractive summary methods take is to firstly construct a representation of the source text with stop word elimination and sentence segmentation. Secondly to score these sentences based on features such as sentence location and term frequency. Lastly a number of these sentences are selected to make a summary. [4], [11]

Preprocessing
Many summarization methods use stop word lists and removes these words before further processing of the text. Stop words are commonly used words that help to build sentences but does not carry any significance themselves like conjunctions and prepositions e.g. “the”, “and” or “on”. These are especially important to remove when methods use word frequency to determine importance of sentences. In such cases the stop words would most likely be classed as “most important” and thus affect negatively on these methods. [12]

Some summarizing methods measure overlap of words between sentences. The problem that emerges is words that have similar meaning like “cats”, “catlike” and “catty” would not overlap because they are different words. A stemming algorithm is needed to reduce these words to a common form, called the “stem”, in this example it would be “cat”. [13]

Sentence scoring
The selection of sentences to include in a summary is done based on scores assigned to the sentences of a document. This score should represent how well a sentence explains important aspects of the text. [4] There are numerous scoring features proposed in different studies and these can be used separately but are more often used in combination. Meena YK et al. [11] listed 22 different scoring features that can be looked into when scoring sentences, some of the features are term frequency, sentence similarity and sentence length. Term frequency is the sum of the frequency of a term within a text and is often used to find terms that are deemed important in a text and score sentences based on these. Sentence similarity scores the vocabulary overlap between sentences to see how similar these are. Sentence length measures the length of a sentence and adds a weight to the sentence, since sentences which are too long or too short are less desirable to use in a summary. [11]

Sentence selection
In the last step the summarizers will select sentences to combine into the final summary. Many summarizers selects the n most important sentences where the user decides how long the summary should be. Some factors summarizers may look at is if the sentences are too similar to each others or if they cover all topics of the original text. [4]

Existing methods
This section will cover the existing methods and how they work that we choose to compare our summarizer with.

Luhn
This method of summarization is based on the paper written by Luhn [14]. Luhn’s method preprocesses the sentences with stemming and ignores very high frequency words that are often stop words. Luhn’s method uses word frequency as well as the relative positioning of a word within a sentence to calculate the relative significance weight of each sentence. The sentences with the highest significance score are then added to the summary of the text. [14]

Latent Semantic Analysis - LSA
LSA is a method that uses contextual information in the article to extract what words are used together and what words are common in a given sentence. Sentences that share high amount of common words has an increased semantic
similarly. There are three steps for an LSA summarization: the making of matrix representation of input, singular value decomposition and sentence selection.

The matrix representation of input is made using words as rows and sentences as columns. The cells within this matrix represents the importance score of the word and this score is calculated with different methods like simple word frequency, depending on implementation.

The next step is to apply singular value decomposition (SVD) to our matrix. This is a mathematical method to detect patterns of similarities between the words and sentences and creates a reduced dimensional representation of this data.

Finally the desired number of sentences is selected from this SVD representation. [15]

**TextRank**

The TextRank model is built based on the PageRank algorithm which is used to rank the importance of web pages in web search engines. The PageRank algorithm is graph-based and the basic idea is using a recommendation system to measure the importances of pages by looking at how many pages link to them. In the graph representation of a document, the pages are represented as nodes and the links as the edges of the graph. Nodes with a high number of edges pointing to it will get a higher score, and edges from nodes with a higher score will be weighted higher than edges from nodes with lower score.

This same principle is used in TextRank where a sentence will be the node but instead of links the edges are sentence similarity instead. To determine the sentence similarity TextRank inspects the content overlap of sentences by measuring overlapping tokens.

The TextRank model also adds a normalization factor to sentences to avoid giving longer sentences a higher score solely because the increased amount of tokens. [16]

**LexRank**

LexRank works very much like TextRank but differs in that it uses cosine similarity instead of content overlap. Cosine similarity is explained further in the non-redundancy part of the theory section. LexRank then adds a postprocessing step during sentence selection, this step discards lines that are too similar in order to reduce redundancies in the summary. [17]

**SumBasic**

SumBasic is similar to Luhn’s method, as both focuses on word frequency. SumBasic however does not remove stop words but uses weights for sentences to combat this. Each word is given a word frequency probability and these probabilities is used to give the weights to the sentences.

SumBasic selects sentences in sequence to add into the summary until the desired length is achieved. Between each addition it updates word probabilities of all the words included in this sentence. This update step gives SumBasic the ability to select next sentence based on information already in the summary and thus reduces redundancy in a natural way. [18]

**Evaluation**

The evaluation of the summaries produced by the different summarization methods will be entirely focused on intrinsic evaluation methods; specifically text quality evaluation in the form of referential clarity and non-redundancy metrics, as well as content-based evaluation in the form of n-gram matching using ROUGE.

**Referential clarity**

One factor to consider when using extractive summarization techniques is the presence of anaphora - references to another entity, the antecedent, within a text [8].

Mitkov presents the example:

“Computational Linguists from many different countries attended the tutorial. They took extensive notes.” [8]

In Mitkov’s example the antecedent is “Computational Linguists” and the anaphora is “They” [8].

One way the presence of an anaphora could affect the textual quality of a summary made through extractive summarization is if the anaphora is dangling, meaning that the sentence containing the anaphora is included in the summary while the sentence including the antecedent is not [19]. In the previous example this would mean that the sentence “They took extensive notes.” would be included in the summary, while the sentence “Computational Linguists from many different countries attended the tutorial.” would not, confusing the reader as to which entity the anaphora “They” is referring to.

**Non-redundancy**

Lloret et al. [9] bring up a well-known problem within the field of automatic text summarization, namely redundancy. Redundancy in the context of text summarization is giving a specific piece of information more than once within the summary, which they in turn claim affects the overall quality of the summary.

In this paper we will use the cosine similarity of the sentences in our summaries to determine the presence of redundancies in our summaries, as Newman et al. [20] found that cosine similarity performed as well as, or better than, other redundancy detection techniques on a corpus of limited size.

Lloret et al. present a three step procedure to use cosine similarity to find redundancies within a summary:

1. Split the summary into a set of sentences;
2. Calculate the cosine similarity between one sentence and all of the remaining sentences; and
3. Consider sentences with a cosine similarity score above a certain threshold value to be redundant.

As presented by Lloret et al., given two sentences \( s_1, s_2 \), the formula to find the cosine similarity between the two sentences in feature vector form is:

\[
sim(s_1, s_2) = \cos(\theta) = \frac{\sum_i s_1(i) s_2(i)}{\sqrt{\sum_i s_1(i)^2 \sum_i s_2(i)^2}}
\]

[9]

The value produced by the cosine similarity formula is a value between 0 and 1, where 0 means the vectors are orthogonal to each other (at 90 degrees to each other, having no matches), whereas a cosine value of close to 1 means that the degree between the two vectors is very small, meaning the sentences are very similar to each other.

[21]

**Grammaticality**

As for the grammatical aspect of textual quality in an extractive summary, Parveen et al. [22] state that the grammaticality of a summary is not a concern when using extractive summarization of complete sentences, since the extracted sentences are assumed to be grammatical. We will therefore not take grammaticality into account when assessing the textual quality of our generated summaries.

**Keywords**

A text can be summarized in multiple ways because a text can contain a lot of information. Two different summarizers can deem different aspects of the text more important. To try to measure how much a summarizer retain important information we can look at *keywords*. The general idea of inspecting keywords in the summaries is the gist that writers use keywords to convey important concepts of the text, and these are usually recurrent. [23]

**ROUGE**

Recall-Oriented Understudy for Gisting Evaluation, or ROUGE for short, is a package created with the purpose to automatically determine the quality of an automatically generated text summary in comparison to a reference gold standard summary. We will make use of three types of ROUGE measurements in this paper; ROUGE-N, ROUGE-L, and ROUGE-W. [10]

ROUGE-N compares n-grams in the automated summary compared to the reference gold standard summaries [10]. N-grams being word sequences, where N determines the number of words in sequence, unigrams being individual words, bigrams being word pairs, trigrams being sequences of three words, and so on.

ROUGE-L measures the *longest common subsequence* (LCS) between a candidate summary and the reference summary. An important distinction from ROUGE-N scoring is that ROUGE-L only considers in-sequence matches, Lin presents an example with the reference sentence “police killed the gunman”, and the two candidate sentences \( S_1 \): “police kill the gunman”, and \( S_2 \): “the gunman kill police”. While both candidate sentences match the words “gunman”, “the” and “police”, the candidate \( S_1 \) will get the ROUGE-L score \( 3/4 \), while \( S_2 \) will get the score \( 2/4 \), since the word “police” appears out of sequence in \( S_2 \).

[10]

ROUGE-W also looks at LCS but this adds weight to it so that a consecutive matching sequence has higher score than a non-consecutive one. To illustrate this difference we can use reference sequence \( R \) and two candidate sequences \( C_1 \) and \( C_2 \):

\[
R: [\text{A}, \text{B}, \text{C}, \text{D}, \text{E}]
\]

\[
C_1: [\text{A}, \text{B}, \text{C}, \text{K}, \text{I}]
\]

\[
C_2: [\text{A}, \text{H}, \text{B}, \text{I}, \text{C}]
\]

In this example both \( C_1 \) and \( C_2 \) would get the same ROUGE-L score because they have the same length on the LCS, but \( C_1 \) would get a higher ROUGE-W score because of its consecutive matches. [10]

As of version 1.5.1\(^1\) ROUGE measures the precision, recall and F1-values of summaries compared to a gold standard summary. In the context of ROUGE-N scores, precision is the fraction of n-grams in the generated summary that also appear in the reference summary, recall is the fraction of n-grams in the reference summary that overlap in the generated summary [24], and the F1 score is calculated as:

\[
F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

[25]

**METHOD**

This section describes in depth how the results are acquired and how these are then evaluated.

**Dataset**

The dataset consists of 100 news articles with accompanying summaries taken from the Swedish news site Kvartal\(^2\). The articles were chosen from Kvartal were published during the period 25/4-2016 - 12/5-2019, only the articles with published summaries over 100 words long were chosen. The article length varies from 1147 words to

---

\(^1\) https://github.com/summanlp/evaluation/tree/master/ROUGE-RELEAS-1.5.5

\(^2\) https://kvartal.se/
6888 words, with an average length of 150 sentences and 3047 words. The length of the summaries varied between 100 and 356 words, with an average of 8 sentences and 155 words, or 5% of the length of the original articles.

While creating our dataset we added punctuation to the end of all headlines without it, this was done in order to simplify the sentence segmentation process. Headlines consisting of a number and nothing else, e.g. “4.”, were removed altogether from the article. We also excluded all text present within blockquote HTML tags, since those quotes were all taken from the article text itself for stylistic purposes, doubling the presence of certain sentences. Next we removed the leading “/” at the beginning of each paragraph of the gold standard summaries, since they were presumably added as a stylistic choice and had no relation to the text of the original article. For the final step we removed any fact boxes, links, images and image texts from the article.

**Data preprocessing**

The preprocessing of the article data in our own preprocessing solution was made in three steps; sentence segmentation, removal of stop words, and finally stemming of the remaining words. These preprocessing steps were only made for the summarization technique we call Modified TextRank, the other summarization models use their own methods for preprocessing the data.

**Sentence segmentation**

Sentence segmentation involves dividing a document into separate sentences, so that they can be processed individually, to accomplish this we used NLTK (Natural Language Toolkit) [26] and its implementation of a sentence tokenizer, turning the articles into lists of sentences.

**Stop word removal**

Each article, now a list of sentence strings, was then processed sentence by sentence to remove all stop words. To identify which words to exclude from the sentences we used a list of Swedish stop words from NLTK [26] in conjunction with a Swedish stop word list created by iMatrics.

**Stemming**

For stemming the sentences we used the Swedish version of the Snowball⁵ stammer in conjunction with two lists containing words and their corresponding stem words, developed by iMatrics.

**Summarization**

To determine the appropriate length of each automatic summary we used a sentence tokenizer to get the number of sentences in each gold standard summary and then generated automatic summaries with the same amount of sentences as the gold standard.

**Scoring**

The scoring of all sentences in Modified TextRank is done using the NetworkX [27] package implementation of the PageRank algorithm.

**Selection of sentences**

After the scoring but before selecting sentences using Modified TextRank, extracts the top ranked sentences in the article, top ranked being defined as the sum of the desired number of sentences for the final summary, the number of sentences in the lead paragraph, and the number of subheadings in the article. The model then removes any headlines that may have ended up in the top sentences.

During the next step the model calculates the cosine similarity between all remaining top ranked sentences and removes the lower ranked sentence if the cosine similarity is greater than or equal to 0.7, the same threshold value as used by Lloret et al. [9] Both this step and the removal of the headlines is done to reduce the redundancy of the generated summary.

Finally the n highest scoring sentences are selected, where n is the desired number of sentences for the summary, and then sorted to the order they originally appeared in the article.

**Comparative summarizers**

The summarizers used for comparisons are all implementations taken from the sumy⁴ library, with the Swedish NLTK [26] stop word list provided to the stop word parameter, the exception being the Luhn summarizer implementation.

**Evaluation**

To evaluate the quality of the summarization methods we looked at four aspects of the resulting summaries: referential clarity within the summaries, non-redundancy, keyword preservation, and content overlap compared to gold standard summaries, in the form of ROUGE scores.

**Referential clarity**

To evaluate the referential clarity of each summarizer we manually read through each summary of 27 articles twice, marking each instance of dangling anaphora and then summed the total amount of dangling anaphoras across all article summaries for each summarizer. The anaphora we counted were; “han”, “hans”, “hon”, “hennes”, “hen”, “de”, “dem” and “deras”. Other words which can function as anaphora such as “den”, “denna”, “det”, “detta” and “dessa” were not counted as dangling anaphora in our evaluation.

---

³ https://snowballstem.org/

⁴ https://github.com/miso-belica/sumy
Non-redundancy
To evaluate the redundancy of each summary we followed the earlier mentioned three step procedure presented by Lloret et al. of splitting each summary into a set of sentences, calculating the cosine similarity between one sentence and each of the remaining sentences in the summary, and then considered the sentences with a cosine similarity over the threshold value to be a redundant sentence [9]. We then inspected all of the sentences with a cosine similarity over our chosen threshold value of 0.7 to make sure that there were no false positives, meaning no sentences marked as redundant that where actually non-redundant.

Keyword preservation
The evaluation of keyword preservation was a four step process using a keyword extractor developed by iMatrics, this program extracts topics like “school” and entities like “Silvia” from a text. The four evaluation steps were:

1. Extract 7 topics and 3 entities as keywords from the original articles;
2. Extract the same amount of keywords from each automatic summary;
3. For each list of keywords extracted from a summary, calculate the recall on the keywords extracted from the original article;
4. Calculate the average recall of keywords for all summarization methods.

ROUGE
The evaluation of the ROUGE scores for each summarization method was made using the py-rouge\(^3\) library. Using the py-rouge library, and the Kvartal summaries as the gold standard summaries, we calculated the precision, recall and F1 values for six different ROUGE categories each automatically generated summary; ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE-L and ROUGE-W. Using the ROUGE scores for all summaries we then calculated the average ROUGE precision, recall and F1 scores across the whole dataset for each summarization technique.

RESULTS
The results will be presented in two sections text quality evaluation and content evaluation.

Text quality evaluation results
The text quality results is further divided into two sections; referential clarity, the data presented in Table 2, and non-redundancy, with the data presented in Table 4.

Referential clarity
As shown in Table 2, SumBasic has the least amount of dangling anaphora among the summarization techniques with 1 instance, while the rest vary between 3 and 4 each, the exception being LSA with 8 dangling anaphora.

<table>
<thead>
<tr>
<th></th>
<th>Modified TextRank</th>
<th>TextRank</th>
<th>LexRank</th>
<th>LSA</th>
<th>Luhn</th>
<th>SumBasic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUGE-1</td>
<td>0.35</td>
<td>0.34</td>
<td>0.35</td>
<td>0.31</td>
<td>0.33</td>
<td>0.37</td>
</tr>
<tr>
<td>ROUGE-2</td>
<td>0.09</td>
<td>0.07</td>
<td>0.08</td>
<td>0.06</td>
<td>0.07</td>
<td>0.07</td>
</tr>
<tr>
<td>ROUGE-3</td>
<td>0.04</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.02</td>
</tr>
<tr>
<td>ROUGE-4</td>
<td>0.03</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>ROUGE-L</td>
<td>0.23</td>
<td>0.23</td>
<td>0.23</td>
<td>0.21</td>
<td>0.22</td>
<td>0.23</td>
</tr>
<tr>
<td>ROUGE-W</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
<td>0.11</td>
<td>0.11</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Table 1. ROUGE precision values. Highest values highlighted.

\(^3\) https://pypi.org/project/py-rouge/

<table>
<thead>
<tr>
<th></th>
<th>Number of dangling anaphoras</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modified TextRank</td>
<td>4</td>
</tr>
<tr>
<td>TextRank</td>
<td>3</td>
</tr>
<tr>
<td>LexRank</td>
<td>3</td>
</tr>
<tr>
<td>LSA</td>
<td>8</td>
</tr>
<tr>
<td>Luhn</td>
<td>4</td>
</tr>
<tr>
<td>SumBasic</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2. Comparison of total amount of dangling anaphora.
Non-redundancy
The result from non-redundancy we got showed us that the unmodified TextRank model yielded greater number of redundant sentences compared to the other methods, followed by Luhn. Textrank had 13 instances redundant sentences, Luhn had 5 redundant sentences whereas the second highest methods TextRank, LexRank and LSA had 2 redundant sentences each. SumBasic and Modified TextRank both had 0 instances of redundant lines scoring above the threshold, as seen in Table 4.

ROUGE precision
The ROUGE-W score between all 6 summarization methods only differed by one percentage point, with 4 methods tied at 0.12, while LSA and Luhn both scored 0.11. The ROUGE-L scores were also fairly evenly distributed, with 4 methods scoring 0.23, Luhn scoring 0.22, and LSA scoring 0.21. The ROUGE-1 scores had a greater spread with SumBasic getting the highest score of 0.37 and LSA getting the lowest at 0.31. The Modified TextRank had the highest scores in the ROUGE-2, ROUGE-3 and ROUGE-4 categories by a margin of one percentage point compared to the others, as well as the tied best result in ROUGE-L and ROUGE-W, as can be seen in Table 1.

ROUGE recall
All six summarization methods tied in the ROUGE-W category with a score of 0.04, the ROUGE-L scores were tied among 4 methods with a score of 0.22, followed by LSA with 0.21, and SumBasic with a score of 0.19. The Modified TextRank had the highest score in 3 categories, ROUGE-2, ROUGE-3 and ROUGE-4, with 0.08, 0.04 and 0.03 respectively, and is tied for the highest score in the ROUGE-1, ROUGE-W and ROUGE-L categories, as seen in Table 3. TextRank had the tied highest scores in the ROUGE-1, ROUGE-W and ROUGE-L

ROUGE F1-value
Five methods had the tied highest score in the ROUGE-W category at 0.06, with SumBasic scoring at one percentage point lower. The Modified TextRank model had the tied highest score with the TextRank model in the ROUGE-L category with a score of 0.23, and the highest scores in the ROUGE-1, ROUGE-2, ROUGE-3 and ROUGE-4 categories at 0.35, 0.08, 0.04, and 0.03. The TextRank

---

<table>
<thead>
<tr>
<th></th>
<th>Modified TextRank</th>
<th>TextRank</th>
<th>LexRank</th>
<th>LSA</th>
<th>Luhn</th>
<th>SumBasic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUGE-1</td>
<td>0.34</td>
<td>0.34</td>
<td>0.33</td>
<td>0.30</td>
<td>0.33</td>
<td>0.29</td>
</tr>
<tr>
<td>ROUGE-2</td>
<td>0.08</td>
<td>0.07</td>
<td>0.07</td>
<td>0.06</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>ROUGE-3</td>
<td>0.04</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>ROUGE-4</td>
<td>0.03</td>
<td>0.02</td>
<td>0.01</td>
<td>0.02</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>ROUGE-L</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
<td>0.21</td>
<td>0.22</td>
<td>0.19</td>
</tr>
<tr>
<td>ROUGE-W</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Table 3. ROUGE recall values. Highest values highlighted.

<table>
<thead>
<tr>
<th></th>
<th>Number of redundant sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modified TextRank</td>
<td>0</td>
</tr>
<tr>
<td>TextRank</td>
<td>13</td>
</tr>
<tr>
<td>LexRank</td>
<td>2</td>
</tr>
<tr>
<td>LSA</td>
<td>2</td>
</tr>
<tr>
<td>Luhn</td>
<td>5</td>
</tr>
<tr>
<td>SumBasic</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4. Comparison of total amount of redundant sentences.

Content evaluation results
The content evaluation results are divided into four categories with one table each, ROUGE precision in Table 1, ROUGE recall in Table 3, ROUGE F1 in Table 5, and keyword preservation in Table 6.
model tied for the best result in the ROUGE-L and ROUGE-W categories, and tied for second best in the other categories, scoring one percentage point lower than the Modified TextRank, as seen in Table 5.

**Keyword preservation**

The results in Table 6 show us that Luhn and LSA with 37.19% and 34.67% retention respectively had the highest scores whereas SumBasic and LexRank with 27.20% and 27.17% retention respectively had the lowest. Our Modified TextRank retained 30.24% of the keywords in the summaries which is slightly below TextRank that had 30.97% retention.

| Average ratio of preserved keywords | 30.24% |
| Modified TextRank | 30.24% |
| TextRank | 30.97% |
| LexRank | 27.17% |
| LSA | 34.67% |
| Luhn | 37.19% |
| SumBasic | 27.20% |

**Text quality**

The number of dangling anaphora is fairly constant across all summarization methods, the exception being LSA with 8 instances of dangling anaphora. Since no steps are being taken to avoid dangling anaphora in the Modified TextRank model it doesn’t perform any better than the comparative summarization methods. For future considerations a system to identify the antecedent for each anaphora and then substituting the anaphora for the antecedent if the sentence containing the antecedent gets excluded from the summary could help avoid dangling anaphora within summaries generated by the model.

The results from measuring the non-redundancy of the summaries showed that Modified TextRank had no redundant sentences and thus performed as good as SumBasic whereas the rest of the methods had a few redundant sentences each, with the exception of TextRank which had 14 instances of redundant sentences, the vast majority being instances where the same, or very similar, sentence was picked both from the main article text and either the lead paragraph or one of the article subheadings. Modified TextRank scoring so well is due to the implementation of a cosine similarity check before the selection of each sentence, where each candidate is compared to all higher scoring sentences up for selection. SumBasic solves this by updating probabilities between selection of sentences. When checking for redundancies using cosine similarity the same threshold value as used by Lloret et al. [9], 0.7, was chosen in order to minimize the risk of non-redundant sentences being considered redundant. As a consequence sentences right below the threshold, which could be considered redundant by a human, did not get marked as such, resulting in false negatives. Similarly when including a cosine similarity check within a summarization model, if the threshold value is set too low it risks incorrectly eliminating non-redundant sentences.

The Discussion

The Modified TextRank performs on par or better than the comparative summarization techniques within both the text quality and content evaluation categories.

---

**Table 5. ROUGE F1 values. Highest values highlighted.**

<table>
<thead>
<tr>
<th></th>
<th>Modified TextRank</th>
<th>TextRank</th>
<th>LexRank</th>
<th>LSA</th>
<th>Luhn</th>
<th>SumBasic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUGE-1</td>
<td>0.35</td>
<td>0.34</td>
<td>0.34</td>
<td>0.31</td>
<td>0.33</td>
<td>0.32</td>
</tr>
<tr>
<td>ROUGE-2</td>
<td>0.08</td>
<td>0.07</td>
<td>0.07</td>
<td>0.06</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>ROUGE-3</td>
<td>0.04</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>ROUGE-4</td>
<td>0.03</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>ROUGE-L</td>
<td>0.23</td>
<td>0.23</td>
<td>0.22</td>
<td>0.21</td>
<td>0.22</td>
<td>0.21</td>
</tr>
<tr>
<td>ROUGE-W</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.05</td>
</tr>
</tbody>
</table>
high scoring sentences from selection and thus result in reduced summary quality.

As an example to demonstrate the difficulty in choosing a threshold value, when using the threshold value 0.6 the sentences “Alla islamister är muslimer, men alla muslimer är inte islamister.” and “Alla jihadister är islamister, men alla islamister är inte jihadister.” were considered redundant by the measure of cosine similarity, despite not being considered redundant by a human reader. Conversely the two sentences “Begreppet människovärde är i grunden religiöst och bygger på idén om Homo sapiens som Guds avbild.” and “Människovärde är ett religiöst begrepp som bygger på tron att Homo sapiens är Guds avbild.” are obviously redundant sentences to a human reader, but scores a lower cosine similarity value than the previously mentioned two sentences, so they will be considered non-redundant in a system focused on eliminating false positives such as our first example.

Another criticism of investigating non-redundancies by use of cosine similarity, brought up by Lloret et al. [9], is that cosine similarity only analyzes the sentences at a lexical level, and does not consider the semantic and syntactic levels of the language.

Content evaluation
The ROUGE results show that the Modified TextRank model performs as well as, or better than, all other summarization methods used for comparison, except for in the case of the ROUGE-1 precision scores where SumBasic has a higher result. This gives us an indication that the changes done to the stemming and stop words used in the Modified TextRank may have had a small positive effect on the model and encourages further testing.

The results from keyword preservation show us that Modified TextRank model is neither the best nor the worst of the compared methods. Modified TextRank was 0.72 percentage points below TextRank indicates that while Modified TextRank performs better in some fields it does not retain quite as much information as TextRank.

Ethical and social aspects
The use of automatic text summarization can give a larger audience a more efficient way to consume information, one such case is brought up in a study by Grefenstette [28] where text summarization is used in a audio scanning device for the blind.

Source criticism
The sources present in our reference list are academically sourced and we therefore assume that they have gone through peer review or similar measures. There are also five footnotes to URLs present in the text, three of them being URLs to the official website of the package, the fourth being a direct link to the official release documentation, and the fifth being a link to the official webpage of the news site Kvartal.

Further work
To improve referential clarity for summaries produced by the Modified TextRank further work is needed on the detection and handling of dangling anaphora, both on the topic of using referential clarity to get more accurate sentence scores, as well as a way to avoid dangling anaphora through the use of substituting the anaphora with the antecedent it is referring to.

Investigations into, and comparisons with, extractive summarization methods based on recurrent neural networks trained on swedish language data would also give an even better and more comprehensive view of the alternatives and how well the methods perform.

Limitations and threats to Validity
The summaries generated by the summarization methods were chosen to be the same number of sentences as the reference gold standard summaries, and the gold standard summaries were on average 5.9% the amount of sentences of the original article. A dataset where the gold standard summaries have a higher length ratio compared to the article text could yield different results.

While investigating the presence of dangling anaphora in the automatically generated summaries, only the words; “han”, “hans”, “hon”, “hennes”, “hen”, “de”, “dem” and “deras” were considered. Since not all possible words that could grammatically function as anaphora were covered there is a possibility that dangling anaphora remained undiscovered. The limited size of the dataset when investigating anaphora is also worth considering.

When selecting extractive summarization methods for comparison we excluded methods based on recurrent neural network due to hardware- and time limitations.

The size of our dataset can give an indication of the capabilities of the different methods, but to get a more statistically significant result the dataset should be higher than the 100 articles we had.

The inclusion of the lead paragraph as well as subheadings within the text data has a negative effect on some of the comparative summarization methods when it comes to redundancy in the summaries, since it introduces the possibility of the methods selecting the same piece of information from both the lead paragraph as well as the main text of the article.

CONCLUSIONS
In conclusion, based on the results garnered by the tests, we can see indications that improved stemming and stopword capabilities could have a positive result on the TextRank model’s ROUGE scores, and that adding a check for cosine similarity while selecting sentences has a positive effect on
avoiding repeated information if the data includes repeated information through subheadings or lead paragraphs within an article. The modifications seems to result in slightly less information retention compared to TextRank and we also see a need for taking referential clarity into account.

ACKNOWLEDGMENTS
We would like to express special thanks to Christoffer Nilsson for his patient assistance and advice. We would also like to thank the rest of the staff at iMatrics for a great work environment.

REFERENCES
Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtidiga ersättare – under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnt som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/.

Copyright

The publishers will keep this document online on the Internet – or its possible replacement – for a period of 25 years starting from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Niko Lehto, Mikael Sjödin