Enhancing extraction based summarization with outside word space

Christian Smith, Arne Jönsson
Santa Anna IT Research Institute AB
Linköping, Sweden
christian.smith@liu.se, arnjo@ida.liu.se

Abstract

We present results from improving vector space based extraction summarizers. The summarizer uses Random Indexing and Page Rank to extract those sentences whose importance are ranked highest for a document, based on vector similarity. Originally the summarizer used only word vectors based on the words in the document to be summarized. By using a larger word space model the performance of the summarizer was improved. Along with the performance, robustness was improved as random seeds did not affect the summarizer as much as before, making for more predictable results from the summarizer.

1 Introduction

Many persons have, for various reasons, problems assimilating long complex texts. Not only persons with visual impairments or dyslexia, but also, for instance, those having a different mother tongue or persons in need of a quick summary of a text. A tool for automatic summarization of texts from different genres as an aid in reading can thus be useful for many persons and purposes.

Automatic summarization can be done in various ways. A common distinction is extract versus abstract summaries. An extract summary is created by extracting the most important sentences from the original text so that the result is a shorter version of the original text with some information still present, for instance the most important sentences or words. An abstract summary on the other hand is a summary where the text has been broken down and rebuilt as a complete rewrite to convey a general idea of the original text. Furthermore, the summaries can be indicative (for instance only providing keywords as central topics) or informative (content focused) (Firmin and Chrzanowski, 1999). The former might be more usable when a reader needs to decide whether or not the text is interesting to read and the latter when a reader more easily needs to get a grasp of the meaning of a text that is supposed to be read.

In this paper we will examine and try to increase the performance of an automatic extraction-based summarizer. Previously, the summarizer has been functioning without aid of outside corpora or training material. While the performance have been good, some improvements utilizing an outside corpus can be achieved.

The technique behind the summarizer will first be described in more detail, after which some results are presented which indicates that the performance can be enhanced by using outside training material.

2 The word space model

The word space model, or vector space model (Eldén, 2007), is a spatial representation of a word's meaning that can reduce the linguistic variability and capture semantically related concepts by taking into account the positioning of words in a multidimensional space, instead of looking at only shallow linguistic properties. This facilitates the creation of summaries, since the positioning in the word space can be used to evaluate the different passages (words or sentences for instance) in relation to a document with regards to informational and semantic content.

Every word in a given context occupies a specific point in the space and has a vector associated to it that can be used to define its meaning.

Word spaces are constructed according to the distributional hypothesis and the proximity hypothesis. In the distributional hypothesis, words that occur in similar contexts have similar meanings so that a *word* is the sum of its contexts and the *context* is the sum of its words, where the *con-*

text can be defined as the surrounding words or the entire document. The proximity hypothesis states that words close to each other in the word space have similar meaning while those far from each other have dissimilar meaning.

The word space can be constructed from a matrix where text units are columns and the words in all text units are rows in the matrix. A certain entry in the matrix is nonzero iff the word corresponding to the row exists in the text unit represented by the column. The resulting matrix is very large and sparse which makes for the usage of techniques for reducing dimensionality and get a more compact representation. Latent Semantic Analysis is one such technique that, however, can be computationally expensive unless used with alternative algorithms (Gorrell, 2006). Random Indexing (Sahlgren, 2005; Kanerva, 1988) is another dimension reduction technique based on sparse distributed representations that provide an efficient and scalable approximate solution to distributional similarity problems.

3 The summarizer

COGSUM is an extraction based summarizer, using the word space model Random Indexing (RI), c.f. Hassel (2007) and a modified version of PageRank (Brin and Page, 1998).

In Random Indexing context vectors are accumulated based on the occurrence of words in contexts. Random Indexing can be used with any type of linguistic context, is inherently incremental, and does not require a separate dimension reduction phase as for instance Latent Semantic Analysis.

Random Indexing can be described as a two-step operation:

Step 1 A unique d-dimensional *index vector* is assigned and randomly generated to each context (e.g. each document or each word). These index vectors are sparse and high-dimensional. They consist of a small number, ρ, of randomly distributed +1s and -1s, with the rest of the elements of the vectors set to 0.

Step 2 *Context vectors* are produced on-the-fly. As scanning the text, each time a word occurs in a context, that context's d-dimensional index vector is added to the context vector for the word. The context window defines a region of context around each word, and the number of adjacent words in a context window is called the context window size, w. For example, with $w = 2$, i.e. a 2x2 context window, the word o_n is represented by the context window c_m as:

$$c_m = [(o_{n-2})(o_{n-1})o_n(o_{n+1})(o_{n+2})],$$

and the context vector of o_n in c_m would be updated with:

$$C_m =$$

$$R(o_{n-2}) + R(o_{n-1}) + R(o_{n+1}) + R(o_{n+2}),$$

where $R(x)$ is the random index vector of x. This process is repeated every time we observe o_n in our data, adding the corresponding information to its existing context vector C. If the context c_m is encountered again, no new index vector will be generated. Instead the existing index vector for c_m is added to C to produce a new context vector for o_n.

Words are thus represented by d-dimensional context vectors that are effectively the sum of the index vectors of all the contexts in which the word appears.

Additionally, the vectors within the sliding contex window can be weighted according to the distance to the focus word. One example is $2^{(1-distance)}$, or $[0.5, 1, 0, 1, 0.5]$ for a 2x2 context window providing a larger weight for words closest to the focus word (Karlgren and Sahlgren, 2001).

After the creation of word context vectors, the similarity between words could be measured by calculating the cosine angle between their word vectors, by taking the scalar product of the vectors and dividing by their norms such as:

$$cos(x, y) = \frac{x \cdot y}{|x|\,|y|} \quad (1)$$

Random Indexing is useful for acquiring the context vectors of terms, it is however not clear how a bigger context, such as a sentence, could be built from the word vectors. A crude way of creating sentence vectors from word vectors would be to simply summarize the vectors of the words in the sentence after they have been normalized to unit length. However, as the number of words in

a sentence increases, so will the sentence similarity to the mean vector. Comparing sentences or documents in this way using cosine will make for larger similarity just by a larger number of words, regardless of relatedness. To alleviate this problem, the mean document vector is subtracted from each of the sentence's word vectors before summarizing the vectors (Higgins and Burstein, 2007), see Equation 2.

$$\vec{sent}_j = \frac{1}{S}\sum_{i=1}^{S}(\vec{w}_i - \vec{doc}) \quad (2)$$

where S denotes the number of words, w, in sentence j and $\vec{doc}$ is calculated as in Equation 3.

$$\vec{doc} = \frac{1}{N}\sum_{i=1}^{N}\vec{w}_i \quad (3)$$

where N denotes the number of unique words.

Words that are similar to the document vector will come closer to the zero vector, while those dissimilar to the document vector will increase in magnitude. When later summarizing the vectors, those of greater magnitude will have increased impact on the total sentence vector so that common, non-distinct, words not contribute as much to the sentence vector. As this reduces the impact of common non-distinct words, there is essentially no need for a stop word list.

COGSUM also uses the Weighted PageRank algorithm in conjunction to its RI-space to rank the sentences (Chatterjee and Mohan, 2007).

The method of using graph-based ranking algorithms for extracting sentences in summarization purposes was proposed by Mihalcea (2004), who introduce the TextRank model. In graph-based algorithms such as TextRank the text need to be represented as a graph, where each vertex depicts a unit of text and the edges between the units represent a connection between the corresponding text units. Graph-based ranking algorithms may be used to decide the importance of a vertex within a graph, by taking into account global information from the entire graph, rather than from only the local context of the vertices. The ranks are thus recursively computed so that the rank of a vertex depends on all the vertices' ranks. In TextRank, PageRank is used to rank the sentences, although it is noted that other ranking algorithms are possible. PageRank is a graph-based ranking algorithm which originally was used to rank home pages automatically and objectively in the Google search engine (Brin and Page, 1998). In TextRank, for the task of sentence extraction, each sentence in a text is represented as a vertex and the relation between sentences are based on their overlap or "similarity", denoted by Equation 4.

$$Similarity(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{log(|S_i|) + log(|S_j|)} \quad (4)$$

Thus, if a sentence adresses certain concepts, the other sentences that share content will get recommended by that sentence in the recursive fashion provided by PageRank.

To use PageRank and Random Indexing for summaries an undirected fully connected graph is created where a vertex depicts a sentence in the current text and an edge between two different vertices is assigned a weight that depicts how similar these are based on a cosine angle comparison of their meaning vectors, see Figure 1. As it is fully connected, all vertices are connected with each other and all out-links are also considered as in-links.

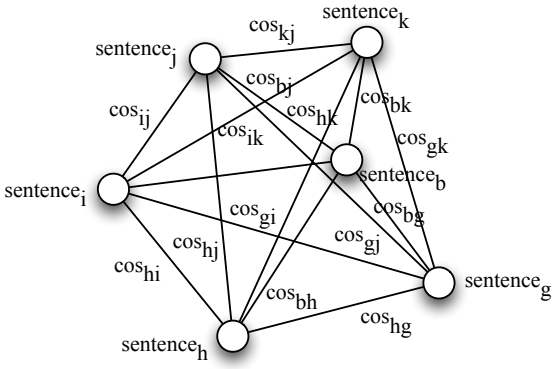

Figure 1: A simplified graph where sentences are linked and weighted according to the cosine values between them.

The algorithm rank ingoing and outgoing links to pages depending on the number of links as follows, Equation 5:

$$PR^W(s_i) = (1-d) + d * \sum_{s_j In(s_i)} w_{ji} \frac{PR^W(s_j)}{\sum_{s_k \in Out(s_j)} w_{kj}} \quad (5)$$

where s_i is the sentence under consideration, $In(s_i)$ is the set of sentences that link to s_i, $Out(s_j)$ is the set of sentences that link from s_i and d is the damping factor.

The damping factor was originally set to account for the possibility of a surfer clicking a random web link when (s)he gets bored (Brin and Page, 1998). With regards to the ranking of sentences, we see the damping factor as the possibility of a sentence containing some implicit information that a certain reader might consider more important at the time, following an analogy by Mihalcea and Tarau (2004). The PageRank-algorithm utilizes the ”random surfer model” and using weighted PageRank in text comparison utilizes ”text surfing” in the context of text cohesion. The links in the sentence graph might be attributed to links between connected concepts or topics semantically, creating a ”web” of understanding on which a reader might surf.

The computation is carried out on all sentences iteratively until node weights converge, see Figure 2.

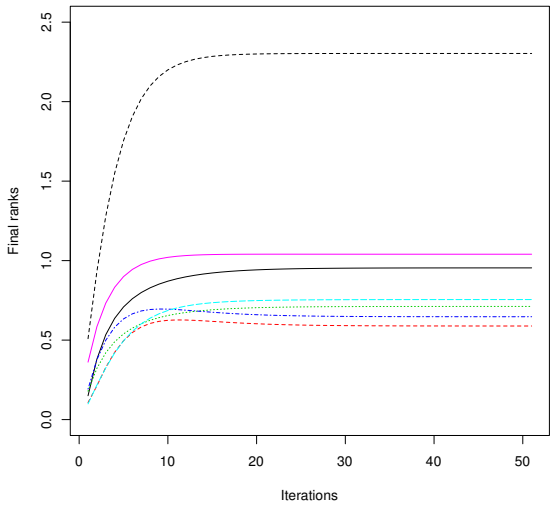

Figure 2: Each line represents a sentence from a single text with its weight plotted on the y-axis on each iteration on the x-axis. 50 iterations are plotted.

The ranking algorithm does not rely only on local context information (vertex) but draws information recursively from the entire graph. Sentences with similar content will then contribute with positive support to each other through a recommendation process, where the sentences' ranks are increased or decreased each iteration. This does not exclusively depend on the number of sentences supporting a sentence, but also on the rank of the linking sentences. This means that a few high-ranked sentences provide bigger support than a greater number of low-ranked sentences. This leads to a ranking of the sentences by their importance to the document at hand and thus to a summary of desired length only including the most important sentences.

When the text has been processed using RI and PageRank, the most important sentences are extracted using the final ranks on the sentences, for instance 30% of the original text, resulting in a condensed version of the original text with the most important information intact, in the form of extracted sentences. Since all sentences are ranked, the length of the summary is easy to specify, in COGSUM this is implemented as a simple slider. COGSUM is designed for informative summaries, but it is also possible to have indicative summaries by clicking a ”keywords” check box.

COGSUM is written in Java and utilizes a Random Indexing toolkit available at Hassel (2011a). No outside material is used which makes the summarizer highly portable and usable for several languages and domains.

Previous evaluations of COGSUM with human users show that summaries produced by COGSUM are useful, considered informative enough and readable (Jönsson et al., 2008). COGSUM has also been evaluated on gold standards for news texts and authority texts showing that it is better than another Swedish summarizer, SweSum, (Dalianis, 2000) on authority texts and almost as good on news texts, texts that the other summarizer was especially adapted to handle (Gustavsson and Jönsson, 2010).

4 Multi-document word vectors

COGSUM has previously worked without aid from any outside source making it highly portable and more or less language independent. However, some problems have been detected. We identified some abruptness in the resulting summaries, affected by the random factor of the index vectors. This was regardless of setting of dimensionality and other parameters.

To investigate the effect of randomness several summaries with different index vectors were created. The final ranks of the sentences in a text after the summarization process were calculated and plotted on each random seed that held its own distribution of the ones in the index vectors, Figure 3. The figure shows 10 different summaries

with their own seeds. The final values after the ranking are plotted on the y-axis mapped to each seed on the x-axis. A straight line would mean that the results are predictable and not affected as much by randomness. As can be seen in Figure 3 there is quite some randomness in which sentences that are chosen depends on the seed to the Random Indexing algorithm.

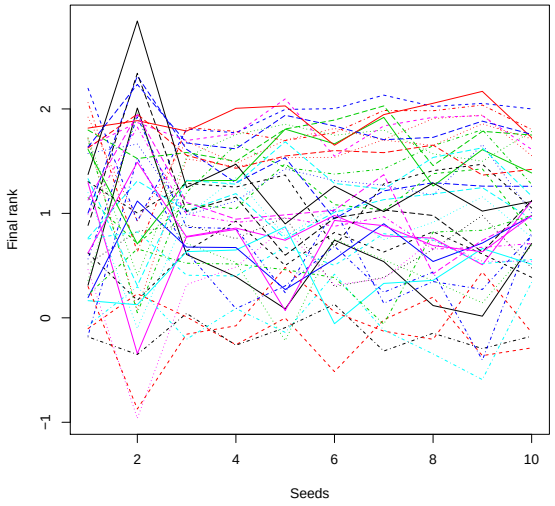

Figure 3: Ten different seeds without pretrained space. Each series represents a sentence in a text. The values on the y-axis are the final values of each sentence after the PageRank-algorithm.

For CogSum, we wanted to extend the method by using an outside larger RI-space. Using a large RI-space, a better semantic representation of the words can be acquired (Sahlgren, 2006). By extending the method to use an outside training space we thus believe that the quality and robustness of the summaries can be improved.

CogSum takes as input the text to be summarized, but now also a previously trained RI-space is supplied, containing the semantic vectors of the words.

The RI-space was created from several Swedish texts from different genres, all in all approximately 240 000 words. The articles consisted of a number of novels in a subset of the Stockholm-Umeå Corpus (Ejerhed et al., 2006), a set of newspaper articles available at the concordances of Språkbanken, specifically from the Parole corpus (Ridings, 2011), and some popular science articles from the same place.

The text is processed by assigning each of the words the corresponding semantic vector from the space. Sentence vectors are constructed as proposed by Chatterjee and Mohan (2007) and Higgins and Burstein (2007), i.e. the words in a sentence are summarized after the subtraction of the mean space vector, and divided by the number of words in the sentence, as in Equation 2.

To investigate the effect on randomness we created 10 summaries with different seeds, the same way as in Figure 3. Figure 4 shows 10 trials on different seeds as before on the same text, but using the larger outside RI-space described above. Comparing figures 3 and 4 reveals a more straight line when using a large RI-space. Thus, by using an outside RI-space, the effect of randomness is reduced and a more predictable result between seeds is achieved.

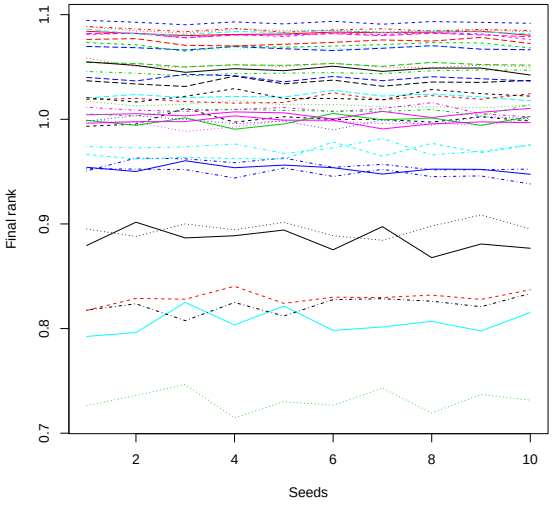

Figure 4: Ten different seeds with pretrained space, each line representing a sentence in a text. The values are the final ranks of each sentence after the PageRank algorithm.

Another problem that has emerged is that the weighted PageRank sometimes fail to converge. This might happen in the original PageRank when the number of outlinks from a vertex is zero (Eldén, 2007). The corresponding phenomenon in the weighted PageRank-algorithm is when the sum of the out weights from a given sentence is zero or close to zero. One reason is that since sentence similarity can be both positive and negative it is possible that they even out. Also, nearly orthogonal sentence vectors makes for weights around zero. This only happens when

not using an outside space. Since a small document does not contain the distribution of words across a large number of context, it is likely that several sentences contain words that occur only in that sentence. When the context vectors for the sentences thereby are created, their vectors may be too sparse and the angle between them becomes nearly orthogonal, and the weights sum up to zero.

Figure 5 is produced similar to Figure 2 , where each line represents a sentence. The ranks in the graph are plotted on the y-axis and each iteration on the x-axis. It is clear that the values fail to converge and stabilize which might be a problem when extracting sentences based on the ranks. This does not happen when using a large RI-space, since the sentence vectors are built from context vectors using a large number of contexts.

The problem can be alleviated simply by redoing the random indexing-phase using a different random seed, which is what is done in the current implementation when not using any outside source.

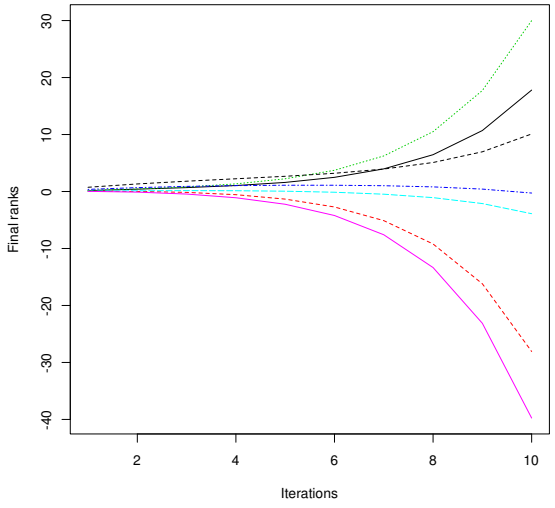

Figure 5: No convergence during the weighted PageRank-algorithm. Each line represents a sentence in a text with the rank plotted on the y-axis and the corresponding iteration during the PageRank on the x-axis. The first ten iterations are plotted.

5 Evaluation

By equipping the summarizer with a better semantic understanding an evaluation was also performed investigating the information quality of the summarizer.

By using a pre-trained RI-space it was hypothesized not only that the random factor could be eliminated, but also that the quality of the summaries would improve. A comparison was made between using an outside text source and using only the document to be summarized to build the RI-space. Since several random seeds provided different summaries on the same text, the average performance measure of 10 seeds for each text was calculated when not using an outside random index space.

The pre-trained RI-space used a dimensionality of 1800, a window size, $w = 2$ with a weighting of $[0.5, 1, 0, 1, 0.5]$, and 8 non-zeroes in the index vectors, similar to Karlgren and Sahlgren (2001).

Using no pre-trained space, the dimensionality was set to 100, window size, $w = 2$ with the same weighting as above, and 4 non-zeroes, to account for a much smaller space, as in Chatterjee and Mohan (2007).

For the evaluation 13 Swedish newspaper articles with a length ranging from 100 to 800 words, see Table 1, were summarized to 30% and compared to human created gold standard summaries of the same length, available at KTHeXtractCorpus (Hassel, 2011b).

Several automatic evaluation packages are available, most notably ROUGE (Lin, 2004). We used, however, also the more recent package AutoSummENG (Giannakopoulos et al., 2008) since it is reported as having a higher correlation with human evaluations than ROUGE (Giannakopoulos, 2009). For AutoSummENG, the comparison was performed by means of graph-value similarity taking content similarity between different texts on character level into consideration. The texts are represented as graphs where each vertex depicts a character n-gram. The graphs from the model and system summaries are then compared resulting in a similarity measure denoting the performance of the system.

It should be noted that no preprocessing in terms of stop word removal and stemming were performed during the ROUGE evaluation since the package is tuned for English and no Swedish lexicon for that purpose were available at the time.

Table 2 shows the values acquired using AutoSummENG for each text and we see that for most texts the summaries produced using the larger RI-space are better than the ones without RI-space.

	Words	Sentences
Text1	110	7
Text2	688	40
Text3	701	37
Text4	400	27
Text5	227	13
Text6	153	9
Text7	441	24
Text8	179	10
Text9	483	33
Text10	838	67
Text11	388	24
Text12	169	9
Text13	471	32

Table 1: Text characteristics, the number of words and sentences on each text.

Using no space, the mean value from all texts of the comparison was 0.420. By using an outside space, the mean value was 0.547 which is a significant improvement ($p < .05$).

As a comparison, evaluations using the more known ROUGE package was performed. When using ROUGE a similar result is obtained, see table 3. Comparing the results of AutoSummENG and ROUGE yields a correlation of $\approx .96$.

6 Conclusion

By using a large word space model the performance of the extraction based summarizer COGSUM could be improved. Along with the performance, robustness was improved, as the random factor between seeds was reduced, making for more predictable results from the summarizer. The performance was evaluated using AutoSummENG, a tool to compare generated texts with gold standard texts created by humans. The evaluation was performed without input from humans, although humans created the gold standard and thus affected the results indirectly, no measures regarding readability were taken. Thus, the measure does not capture readability, only that the extracted sentences can be seen as the most important for the document and that they correspond to human created gold standards.

Evaluations were also performed using ROUGE to have a point of reference since AutoSummENG is a lesser known method of evaluation and the results from these two different packages correlated strongly.

AutoSummEnG	Without space	With space
Text1	0.301	0.751
Text2	0.484	0.484
Text3	0.497	0.509
Text4	0.569	0.447
Text5	0.276	0.556
Text6	0.321	0.520
Text7	0.510	0.502
Text8	0.239	1.000
Text9	0.340	0.465
Text10	0.347	0.419
Text11	0.487	0.556
Text12	0.574	0.384
Text13	0.520	0.517
Mean	**0.420**	**0.547**

Table 2: Evaluation of each summary. Each summary has been compared to a gold standard created by humans. The left column shows the values acquired for the summaries using no outside random indexing-space and the right column shows the values after using an outside space. The values are acquired by means of graph value similarity using AutoSummENG.

ROUGE-1	Without space	With space
Text1	0.386	0.695
Text2	0.538	0.551
Text3	0.570	0.540
Text4	0.647	0.522
Text5	0.290	0.590
Text6	0.368	0.599
Text7	0.600	0.573
Text8	0.359	0.975
Text9	0.452	0.541
Text10	0.454	0.560
Text11	0.574	0.665
Text12	0.682	0.369
Text13	0.599	0.625
Mean	**0.502**	**0.600**

Table 3: Evaluation of each summary using ROUGE-1 n-gram. Each summary has been compared to a gold standard created by humans. The left column shows the ROUGE scores acquired for the summaries using no outside random indexing-space and the right column shows the scores after using an outside space.

Further improvements can be seen with regards to stabilizing weights in the weighted PageRank-algorithm. By using a large word space the sentence vectors become more dense since they are built from context vectors from a large number of contexts. The sentence vectors are thus not as likely to be nearly orthogonal which becomes a problem when summarizing the weights as outlinks, since the sum then might be close to zero.

An increased quality in semantic representation however comes with some tradeoffs. A large word space reduces the portability somewhat, and increases the computational effort since a large space uses a much larger dimensionality. Also, the word space makes it language dependent, a previously strong argument for this method. Creating a larger RI-space for a new language is, however, not such a difficult task if a large enough corpus is available.

The word space that was used was produced from rather general texts and it would be interesting for the future to investigate the effect of different RI-spaces on different genres and domains, both in terms of training material but also on the quality of the summaries. Since Random Indexing is incremental, it is easy to add documents to the semantic space.

Although previous work (Smith and Jönsson, 2011) have looked at readability and concluded that the readability may be increased using extraction based summarization, it is still unclear exactly how cohesive they are. Mihalcea and Tarau (2004) draws an analogy between the PageRank "random surfer model" and "text surfing" which relates to the concept of text cohesion. The links in the graph might be attributed to links between connected concepts or topics in a semantic way so it would not be surprising to find that the summaries have acceptable cohesion. Future research will have to conclude the cohesiveness of the summaries and how they may need to be improved.

We have, however, shown that the quality and robustness can be improved by using an outside previously trained random indexing space in the process of vector space model extraction based automatic summarization.

Acknowledgments

This research was partly supported by a research grant from The Swedish Post and Telecom Agency (PTS).

References

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.

Nilhadri Chatterjee and Shiwali Mohan. 2007. Extraction-based single-document summarization using random indexing. In *Proceedings of the 19th IEEE international Conference on Tools with Artificial intelligence – (ICTAI 2007)*, pages 448–455.

Hercules Dalianis. 2000. Swesum – a text summarizer for swedish. Technical Report TRITA-NA-P0015, IPLab-174, NADA, KTH, Sweden.

Eva Ejerhed, Gunnel Källgren, and Benny Brodda. 2006. Stockholm umeå corpus version 2.0.

Lars Eldén. 2007. *Matrix Methods in Data Mining and Pattern Recognition*. Society for Industrial & Applied Mathematics (SIAM).

Thérese Firmin and Michael J Chrzanowski, 1999. *An Evaluation of Automatic Text Summarization Systems*, volume 6073, pages 325–336. SPIE.

George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. 2008. Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech Language Processing*, 5(3):1–39.

George Giannakopoulos. 2009. *Automatic Summarization from Multiple Documents*. Ph.D. thesis, University of the Aegean.

Genevieve Gorrell. 2006. *Generalized Hebbian Algorithm for Dimensionality Reduction in Natural Language Processing*. Ph.D. thesis, Linköping University.

Pär Gustavsson and Arne Jönsson. 2010. Text summarization using random indexing and pagerank. In *Proceedings of the third Swedish Language Technology Conference (SLTC-2010), Linköping, Sweden*.

Martin Hassel. 2007. *Resource Lean and Portable Automatic Text Summarization*. Ph.D. thesis, ISRN-KTH/CSC/A–07/09-SE, KTH, Sweden.

Martin Hassel. 2011a. Java random indexing toolkit, January 2011. `http://www.csc.kth.se/~xmartin/java/`.

Martin Hassel. 2011b. Kth extract corpus (kthxc), January 2011. `http://www.nada.kth.se/~xmartin/`.

Derrick Higgins and Jill Burstein. 2007. Sentence similarity measures for essay coherence. In *Proceedings of the 7th International Workshop on Computational Semantics (IWCS), Tilburg, The Netherlands*.

Arne Jönsson, Mimi Axelsson, Erica Bergenholm, Bertil Carlsson, Gro Dahlbom, Pär Gustavsson, Jonas Rybing, and Christian Smith. 2008. Skim reading of audio information. In *Proceedings of the The second Swedish Language Technology Conference (SLTC-08), Stockholm, Sweden*.

Pentii Kanerva. 1988. *Sparse distributed memory*. Cambridge MA: The MIT Press.

Jussi Karlgren and Magnus Sahlgren. 2001. From words to understanding. In Y. Uesaka, P.Kanerva, and H. Asoh, editors, *Foundations of Real-World Intelligence*, chapter 26, pages 294–308. Stanford: CSLI Publications.

Chin-yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *ACL Text Summarization Workshop*, pages 25–26.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain.

Rada Mihalcea. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, ACLdemo '04, Morristown, NJ, USA. Association for Computational Linguistics.

Daniel Ridings. 2011. Parole corpus at språkbanken. `http://spraakbanken.gu.se/parole/`.

Magnus Sahlgren. 2005. An Introduction to Random Indexing. *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*.

Magnus Sahlgren. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.* Ph.D. thesis, Stockholm University, Department of Linguistics.

Christian Smith and Arne Jönsson. 2011. Automatic summarization as means of simplifying texts, an evaluation for swedish. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NoDaLiDa-2010), Riga, Latvia*.