Towards Holistic Summarization – Selecting Summaries, Not Sentences

Martin Hassel & Jonas Sjöbergh

KTH CSC
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
{xmartin,jsh}@kth.se

Abstract

In this paper we present a novel method for automatic text summarization through text extraction, using computational semantics. The new idea is to view all the extracted text as a whole and compute a score for the total impact of the summary, instead of ranking for instance individual sentences. A greedy search strategy is used to search through the space of possible summaries and select the summary with the highest score of those found. The aim has been to construct a summarizer that can be quickly assembled, with the use of only a very few basic language tools, for languages that lack large amounts of structured or annotated data or advanced tools for linguistic processing. The proposed method is largely language independent, though we only evaluate it on English in this paper, using ROUGE-scores on texts from among others the DUC 2004 task 2. On this task our method performs better than several of the systems evaluated there, but worse than the best systems.

1. Introduction

Automatic text summarization is a technique where a computer automatically creates an abstract, or summary, of one or more texts. The initial interest in automatic shortening of texts was spawned during the sixties in American research libraries. Since then the technique has been developed for many years (Luhn, 1958; Edmundson, 1969; Salton, 1988) and in recent years, with the current explosion of digital data readily available on public and corporate networks, we have seen an awakening interest for summarization techniques. Today, with digitally stored information available in abundance and in a myriad of forms to an extent as to making it near impossible to manually search, sift and choose which information one should incorporate, this information must instead be filtered and extracted in order to avoid drowning in it. Automatic text summarization is often seen as an important part of this information managing process.

1.1. Summarization Approaches

Summarization approaches are often divided into two groups, text abstraction and text extraction. Text abstraction, being the more challenging task, is often meant as to parse the original text in a deep linguistic way, interpret the text semantically into a formal representation. From this representation new more concise concepts to describe the contents of the text are derived and used to generate a new shorter text, an abstract, with the same information content.

Text extraction, on the other hand, means to identify the most relevant passages in one or more documents, often using statistically based information retrieval techniques augmented with more or less shallow natural language processing and heuristics. These passages, often sentences or phrases, are then extracted and pasted together to form a summary that is shorter than the original document with as little redundancy and information loss as possible. Sometimes the extracted fragments are post-edited or re-written, for example by deleting subordinate clauses or joining incomplete clauses to form complete clauses (Jing and McKeown, 2000; Jing, 2000), thus in some sense approaching abstraction. In this paper we will operate solely within the text extraction paradigm.

1.2. Extractive Summarization

Any attempt at automatic summarization has to address the problem of choosing relevant information from the original document(s) to include in the summary. In extraction-based summarization systems this is most often accomplished by ranking individual segments (sentences, paragraphs, etc.) in the text(s) being summarized (Carbonell and Goldstein, 1998; Hovy and Lin, 1999; Dalianis, 2000; McDonald and Chen, 2002). A typical extractive system selects sentences for inclusion in the summary one at a time, with later choices sensitive to their similarity to earlier ones; the selected sentences are then ordered either chronologically or by relevance. Most extractive systems do not view and weigh the summary, or potential summaries, as a whole in regards to the original document, without a strong discriminatory focus on its parts.

1.3. Summarization Using Word Spaces

Word space models, most notably Latent Semantic Analysis/Indexing (Landauer et al., 1998), enjoy considerable attention in current research on computational semantics. Since its introduction in 1990 LSA has more or less spawned an entire research field with a wide range of word space models as a result, and numerous publications reporting exceptional results in many different tasks, such as information retrieval, various semantic knowledge tests (for example the TOEFL test), text categorization and word sense disambiguation.

The general idea behind word space models is to use statistics on word distributions in order to generate a high-dimensional vector space. In this vector space the words are represented by context vectors whose relative directions are assumed to indicate semantic similarity. The basis of this

assumption is the *distributional hypothesis*, according to which words that occur in similar contexts also tend to have similar properties (meanings/functions). From this follows that if we repeatedly observe two words in the same (or very similar) contexts, then it is not too far fetched to assume that they also mean similar things (Sahlgren, 2005).

Many experiments have been carried out and more than one system has been constructed that attempts to benefit from LSA for the reduction of redundancy within the summary. This is usually accomplished by semantically comparing highly ranked sentences and disfavoring close matches (Gong and Liu, 2001; Ravindra et al., 2004). Also, experiments using LSA as a means of summarization evaluation have been carried out with the positive effects that all co-ordinates of the vectors contribute when calculating the document-summary similarity, and that the effect of near synonymy is reduced (Donaway et al., 2000). A seemingly appealing approach should thus be to utilize this *during* the summarization process in order to try to devise a summary that is in essence a near equivalent to the main topics of the original document, only shorter.

2. Random Indexing

The aim has been to construct a summarizer that can be quickly assembled, with the use of only a few basic language tools, for languages that lack large amounts of structured or annotated data or advanced tools for linguistic processing. In our experiments we have employed the Random Indexing word space approach (Sahlgren, 2001; Sahlgren, 2005), which presents an efficient, scalable and inherently incremental alternative to standard word space methods. As an alternative to LSA-like models that first construct a huge co-occurrence matrix and then use a separate dimension reduction phase, Random Indexing instead accumulates context vectors on-the-fly based on the occurrence of words (tokens) in contexts, without a need for a separate dimension reduction phase.

This technique can readily be used with any type of linguistic context and can be used to index using a more traditional bag-of-tokens approach as well as using a sliding context window (i.e. co-occurrence between words) capturing sequential relations between tokens. These tokens can be the word simply represented by its lexical string, its lemma, or more elaborate approaches utilizing tagging, chunking, parsing or other linguistic units can be employed.

As with all LSA-like models Random Indexing needs, for good performance, large amounts of text (millions of words) when generating the conceptual representations. Since Random Indexing is resource lean and only requires access to raw (unannotated) text, this is generally not a problem.

2.1. Building Context Vectors

The construction of context vectors using Random Indexing can be viewed as a two-step process. First, each context in the data is assigned a unique and (usually) randomly generated label. These labels can be viewed as sparse, high-dimensional, and ternary vectors.[1] This means that their dimensionality (d) usually is chosen to be in the range of a couple of hundred up to several thousands, depending of the size and redundancy of the data, and that they consist of a very small number (usually about 1-2%) of randomly distributed +1s and -1s, with the rest of the elements of the vectors set to 0.

Next, the actual context vectors are produced by scanning through the text and each time a token w occurs in a context (e.g. in a document or paragraph, or within a sliding context window), that context's d-dimensional random label is added to the context vector for the token w. Thus, when using a sliding context window, all tokens that appear within the context window contribute (to some degree) with its random label to w's context vector. Words are in this way effectively represented by d-dimensional context vectors that are the sum of the random labels of the co-occurring words. When using a sliding context window it is also common to use some kind of distance weighting in order to give more weight to tokens closer in context.

2.2. Conceptual Representations of Documents

The core of our approach is thus to try to capture the essence of the document being summarized by use of computational semantics. We accomplish this by first building semantic, or conceptual, representations for each word based on a large corpus, in our case the British National Corpus (Burnard, 1995), as well as the documents themselves. One of the strengths of Random Indexing is that when building the representation for the document about to be summarized, we can in a very elegant way fold the document currently being summarized into the Random Index, immediately taking advantage of (possibly genre or text type specific) distributional patterns within the current document. Apart from the advantage of eliminating the risk of lack of data due to unknown words, we also have a system that learns more as it summarizes. The problem of sparse data cannot be completely avoided, since a never before seen word will only have as many contextual updates as the number of times it occurs in the document. This is however far better than no updates at all.

Since the random labels are very sparse high dimensional vectors they are approximately orthogonal. This means that if we collect the context vectors produced by Random Indexing in a matrix, this matrix will be an approximation of the standard co-occurrence matrix in the sense that their corresponding rows are similar or dissimilar to the same degree. In this way, we can achieve the same dimensional reduction as is done in LSA by use of SVD; transforming the original co-occurrence counts into a much smaller and denser representation (Sahlgren, 2005). A key factor is thus the stability in the results over different random projections.

3. Our Method in a Nutshell

Our method aims at producing overview summaries. This is accomplished by trying to find a summary of a given length

[1] The extremely sparse random labels are handled internally as short lists of positions for non-zero elements, and are generated on-the-fly whenever a never before seen token is encountered in the context during indexing.

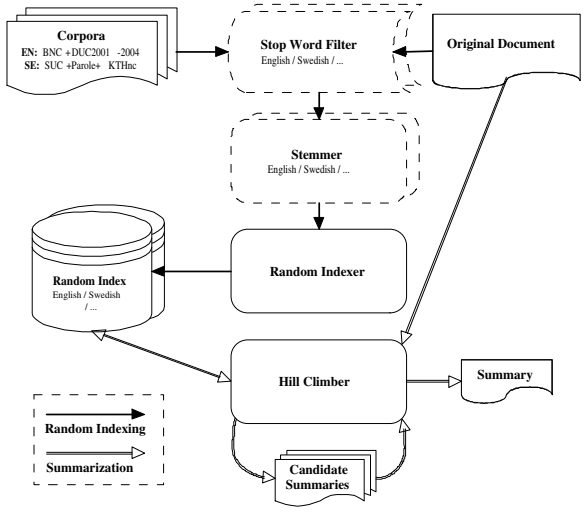

Figure 1: HolSum system layout. The candidate summaries are iterativley generated and evaluated (i.e. compared for document similarity against the original document). Stopword filtering and stemming are optional but have shown to improve the result.

that is as similar to the original text as possible. One way to accomplish this would be to generate all possible extracts and see which one is most similar to the original text. Besides being computationally cumbersome, the difficulty here lies in judging how similar two texts are. Most methods that compare two documents use measures like word or n-gram overlap. Since all candidate summaries here are extracts from the original text, all words in all summaries overlap with the original text. This is thus not a good way to differentiate between different candidates.

3.1. Evaluating Candidate Summaries

Our method makes use of Random Indexing to differentiate between different summaries. As discussed in section 2., Random Indexing gives each word a context vector that in some sense represents the semantic content of the word. We make use of these vectors when calculating a measure of similarity between two texts.

Each text is assigned its own vector for semantic content, which is simply the (weighted) sum of all the context vectors of the words in the text. This can be seen as projecting the texts into a high dimensional vector space where we can relate the texts to each other. Similarity between two texts is then measured as the similarity between the directions of the semantic vectors of the texts, in our case between the vector for the full text and the vectors for each of the candidate summaries. Similar approaches have also been applied to for instance text categorization (Sahlgren and Cöster, 2004).

When constructing the semantic vector for a text, the context vector for each word is weighted with the term frequency and the inverse document frequency, by making the length of the vector be `tf*log(idf)`. If desired, other weighting criteria can easily be added, for instance for slanted or query based summaries where some words are deemed more important, or by giving words occurring early in the document, in document or paragraph headings etc. higher weight. In our experiments we have, however, only used `tf*log(idf)`.

Words in a text that have never been encountered during the calculation of a word space representation generally degrade performance. This is not a problem with Random Indexing though, since it allows for continuous updates. Simply add the new text to the index immediately before summarizing.

Since our method does not give any consideration to the position in the text a sentence is taken from (though that is possible to do if one so wishes), it should be relatively straightforward to use for multi-document summarization as well. A system layout can be found in figure 1.

3.2. Finding a Good Summary

To find a good summary we start with one summary and then try to see if there is another summary that is "close" in some sense, that is also a better summary. Better in this context means more similar to the original text. The reason we do not exhaustively pursue the best summary of all possible summaries is that there are exponentially many possible summaries. Comparing all of them to the original text would thus not be feasible.

In our experiments we use the "lead" summary, i.e. the first sentences from the document up to a specified length, as a starting point for our search. Using a standard hill-climbing algorithm we then investigate all neighbours, looking for a better summary. The summaries that are defined as neighbours to a given summary are simply those that can be created by removing one sentence and adding another. Since sentences vary in length we also allow removing two sentences and adding one new, or just adding one new sentence. This allows for optimizing the summary size for the specified compression rate.

When all such summaries have been investigated, the one most similar to the original document is updated to be the currently best candidate and the process is repeated. If no other summary is better than the current candidate, the search is terminated. It is also possible to stop the search at any time if so desired, and return the best candidate so far.

In our experiments on the texts provided for the Document Understanding Conferences (DUC, 2005) the generated summaries are very short, about three sentences. This means that there are usually quite few, typically around four, search iterations, though some documents require very many iterations before a local maximum is found.

Example of a lead summary used as starting point for the greedy search can be found in figure 2. As we can see, the lead summary is just the leading sentences within one document, and as such only covers the aspects of the document chosen to be presented there. Since our method tries to find a summary that is more similar to the view it has of the whole document, it thus transforms the initial summary to a summary with a wider coverage (if no slanting strategies are applied). The local maximum summary corresponding to the lead summary given in figure 2 is presented in figure 3.

```
Azerbaijani President Heydar Aliyev, who is considered the most likely to win the
presidential elections, cast his vote today, Sunday, at one of the polling centers near
his residence in the center of the capital and took the opportunity to attack his main
opponent, Etibar Mammadov.  The president, who was elected in September 1993, said in a
statement to reporters that "one of the candidates, and you know who I mean, asserts that
he has a team and a program, but when the country was on the verge of civil war in 1993,
Etibar Mammadov was involved in the political scene so why did he not do anything and why
did he not try to stop" the tragedy.
```

Figure 2: Lead summary used as starting point for greedy search (ROUGE-1 37.78%, cosine 8.475e-05).

```
Supporters of Azerbaijani President Heydar Aliyev proclaimed today, Monday, that he
was re-elected for a new term from the first round that took place yesterday, Sunday,
while his main opponent Etibar Mammadov, declared that a second round ought to be held.
The 4200 polling offices, under the supervision of 180 observers from the Security and
Cooperation Organization in Europe, will remain open till 20:00 local time.  In order to
win in the first round as Aliyev hopes, a candidate must win more than 75% of the votes
with a turnout of over 25%.
```

Figure 3: Local maximum summary scoring ROUGE-1 43.95% (0.9952 cosine similarity to original document).

4. Evaluation

For reasons of comparability and the benefit of a human ceiling, we have chosen to mimick the evaluation setup for task 2 in DUC 2004 (Over and Yen, 2004). As in this evaluation campaign we have carried out or evaluation using ROUGEeval (Lin, 2003) with the same data and model summaries, both of which will get a brief introduction below.

4.1. Evaluation Data

We have chosen to focus our experiments on the data provided for summarization evaluation campaign during the Document Understanding Conferences. While our method itself is largely language independent, and thus should work comparably well on other languages given enough raw text, the data prepared for the DUC evaluations is widely used and as such forms a basis for comparison to other systems and methods. The evaluation was carried out by first using all manually created 100 word summaries provided for DUC 2004 as reference summaries, trimming our system with different tokenizers and pre-processors (e.g. sentence splitting, stopword filtering, stemming etc.), comparing our results to those reported in (Over and Yen, 2004). Having reached a reasonable level of success we then compared against the complete set of man-made 100 word summaries from DUC 2001-2004 in order to verify our method on a larger test set.

The data used for building the conceptual representations of the documents, as described in section 2.2., is comprised of the British National Corpus (100 million words) as well as roughly 2 million words contained in 291 document sets provided for DUC 2001-2004. After stopword filtering and stemming this resulted in almost 290,000 unique stems taken from 4415 documents.

4.2. Evaluation Metric

The evaluation has been carried out computing ROUGE scores on the system generated summaries using manual summaries provided for DUC as reference, or model summaries. The ROUGE score is a recall-based n-gram co-occurrence scoring metric that measures content similarity by computing the overlap of n-grams occurring in both a system generated summary as well as a set of (usually man-made) model summaries. Throughout the evaluations we have, as in DUC 2004, used ROUGEeval-1.4.2 with the following settings:

```
rouge a -c 95 -b 665 -m -n 4 -w 1.2
```

This means that we use a 95% confidence interval, truncate model and peer at 665 bytes, Porter Stem models and peers and calculate ROUGE-1..4. Also, stopwords are not removed when calculating the score. ROUGE scores have in several studies (Hovy and Lin, 2002; Lin and Hovy, 2003a) proven to correlate highly with human evaluation and has high recall and precision in predicting statistical significance of results comparing with its human counterpart (Lin and Hovy, 2003b).

In our experiments ROUGE scores are in the case of DUC 2004 calculated over 114 system generated summaries, one for each document set, and in the case of DUC 2001-2004 over 291 summaries. A human ceiling (see table 1) has for

	DUC 2004	DUC 2001 - 2004
Human mean	42.6	39.7
Holistic-250	33.9	32.0
Holistic-500	34.2	32.3
Holistic-1000	34.1	32.4
Holistic-RAW	32.7	30.9
Holistic-noRI	30.3	28.5
Baseline-Lead	31.0	28.3

Table 1: ROUGE-1 scores, in %, for different dimensionality choices of the context vectors. RAW indicates no use of stemming and stopword filtering, and noRI uses a traditional `tf*idf` weighted vector space model instead of Random Indexing.

reference been calculated by, for each text set, taking the mean of the ROUGE scores for each man-made summary compared to the remaining man-made summaries (i.e. in turn treating each human-written summary as a system summary). Also, we evaluate a baseline (lead), which is the initial sentences in each text up to the allowed summary length.

5. Results

In the evaluations here we have removed stopwords and used stemming. Two brief evaluations not using these two strategies showed that both approaches result in considerable improvements.

We evaluated the impact of the dimensionality chosen for the Random Indexing method by running our experiments for three different values for the dimensionality, building semantic representations using 250, 500 and 1000 dimensions. Our results show little variation over different dimensionalities. This means that as long as we do not choose too few dimensions, the dimensionality is not a parameter that needs considerable attention.

For each dimensionality we also calculated the mean performance using four different random seeds, since there is a slight variation in how well the method works with different random projections, see section 2.2. The dimensionality showing the most variation in our experiments spanned 33.8-34.4% ROUGE-1. Variations for the other dimensions were slightly less. As shown in table 1, our best run resulted in a mean performance of 34.2%. The best systems in DUC 2004 scored roughly 39% (Over and Yen, 2004). Concerning scores for ROUGE-2..4 our system unsurprisingly follows the pattern of the results reported in the DUC 2004 evaluation campaign, with considerably lower ROUGE-2 (mean 7.2% with 500 dimensions) and almost non-existing scores for ROUGE-3 (mean 2.3%) and ROUGE-4 (mean 1.0%).

Some naïve attempts at sentence compression by removing "uninteresting" text, such as removing anything mentioned within parenthesis were done. We also tried joining sentences together if the second sentence began with `but`, `and`, `however`, `although` or similar text binding markers, indicating that the sentences were in some sense dependent. All such experiments, however, degraded the performance.

6. Conclusions

We have devised and evaluated a novel extraction-based summarization method that, in contrast to most extraction-based systems, does not rank the individual extract segments contributing to the summary. Instead it compares complete summaries to the original text, and chooses the best summary candidate it can find by a simple search strategy.

The method requires no sophisticated tools, though stopword filtering and simple stemming clearly improved the results. Our method is also largely language independent and should work without much modification for multi-document summarization. For good performance access to large amounts of raw (unannotated) text is needed, but for many languages this is readily available.

Intrinsic evaluation has been conducted using ROUGEeval with the man-made summaries from DUC 2001-2004 as model summaries. Evaluation carried out mimicking the DUC 2004 Task 2 set-up places our system in the top half with a ROUGE-1 mean of 34.2%, and also beating the baseline systems by quite a good margin. These results have been verified to be stable also using data provided for DUC 2001-2003. However, there is still room for improvements closing the gap up to the top performing systems and human performance.

7. References

Lou Burnard. 1995. The Users Reference Guide for the British National Corpus.

Jaime G. Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In Alistair Moffat and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336, Melbourne, Australia.

Hercules Dalianis. 2000. SweSum - A Text Summarizer for Swedish. Technical report, KTH NADA, Sweden, October.

Robert L. Donaway, Kevin W. Drummey, and Laura A. Mather. 2000. A Comparison of Rankings Produced by Summarization Evaluation Measures. In Udo Hahn, Chin-Yew Lin, Inderjeet Mani, and Dragomir R. Radev, editors, *Proceedings of the Workshop on Automatic Summarization at the 6th Applied Natural Language Processing Conference and the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, pages 69–78. Association for Computational Linguistics, April 30.

DUC. 2005. Document understanding conferences. http://duc.nist.gov/.

H. P. Edmundson. 1969. New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285, April.

Yihong Gong and Xin Liu. 2001. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, Louisiana.

Eduard Hovy and Chin Yew Lin. 1999. Automated Text Summarization in SUMMARIST. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 81–94. The MIT Press.

Eduard Hovy and Chin-Yew Lin. 2002. Manual and Automatic Evaluation of Summaries. In Udo Hahn and Donna Harman, editors, *Proceedings of the Workshop on Text Summarization at the 4Oth Meeting of the Association for Computational Linguistics*, July 11–12.

Hongyan Jing and Kathleen R. McKeown. 2000. Cut and Paste-Based Text Summarization. In *Proceedings of the 6th Applied Natural Language Processing Conference and the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 178–185, Seattle, Washington, April.

Hongyan Jing. 2000. Sentence Reduction for Automatic Text Summarization. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pages 310–315, Seattle, Washington, April 29–May 4.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284.

Chin-Yew Lin and Eduard Hovy. 2003a. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In Udo Hahn and Donna Harman, editors, *Proceedings of the 2003 Human Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May 27–June 1.

Chin-Yew Lin and Eduard Hovy. 2003b. The potential and limitations of automatic sentence extraction for summarization. In Dragomir Radev and Simone Teufel, editors, *HLT-NAACL 2003 Workshop: Text Summarization (DUC03)*, Edmonton, Alberta, Canada, May 31 - June 1. Association for Computational Linguistics.

Chin-Yew Lin. 2003. ROUGE: Recall-oriented understudy for gisting evaluation. http://www.isi.edu/~cyl/ROUGE/.

Hans Peter Luhn. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research Development*, 2(2):159–165.

Daniel McDonald and Hsinchun Chen. 2002. Using Sentence Selection Heuristics to Rank Text Segments in TXTRACTOR. In *Proceedings of the 2nd ACM/IEEE Joint Conference on Digital Libraries*, pages 25–38, Portland, Oregon.

Paul Over and James Yen. 2004. An introduction to duc 2004 intrinsic evaluation of generic new text summarization systems. http://www-nlpir.nist.gov/projects/duc/pubs/2004slides/duc2004.intro.pdf.

G. Ravindra, N. Balakrishnan, and K. R. Ramakrishnan. 2004. Multi-document automatic text summarization using entropy estimates. In *SOFSEM*, pages 289–300.

Magnus Sahlgren and Rickard Cöster. 2004. Using Bag-of-Concepts to Improve the Performance of Support Vector Machines in Text Categorization. In *Proceedings of the 20th International Conference on Computational Linguistics, COLING 2004*, Geneva, Switzerland, August 23-27.

Magnus Sahlgren. 2001. Vector-Based Semantic Analysis: Representing word meanings based on random labels. In *Proceedings of Semantic Knowledge Acquisition and Categorisation Workshop at ESSLLI'01*, Helsinki, Finland.

Magnus Sahlgren. 2005. An Introduction to Random Indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005)*, Copenhagen, Denmark, August 16.

Gerard Salton. 1988. *Automatic Text Processing*. Addison-Wesley Publishing Company.