SUTLER: Update SummarizER based on Latent Topics

Josef Steinberger[1] and Karel Ježek[1]

[1]Department of Computer Science and Engineering,
Faculty of Applied Sciences,
University of West Bohemia in Pilsen, Czech Republic
jstein@kiv.zcu.cz and jezek_ka@kiv.zcu.cz

Abstract. This paper deals with our past and recent research in text summarization. We went from single-document summarization through multi-document summarization to update summarization. We describe the development of our summarizer which is based on latent semantic analysis (LSA). The classical LSA-based summarization model was improved by Iterative Residual Rescaling. We propose the update summarization component which determines the redundancy and novelty of each topic discovered by LSA. Moreover, we have modified the sentence selection component in order to prevent inner summary redundancy. The results of our first participation in TAC/DUC evaluation seem to be promising.

1 Introduction

Four years ago we started to develop a summarization method whose core was covered by latent semantic analysis (LSA) (*Landauer & Dumais, 1997*). The proposed single-document method (*Steinberger & Ježek, 2004*) modified the first summarization approach, which used LSA representation of a document (*Gong & Liu, 2002*). From single-document summarization we went on to produce multi-document summaries (*Steinberger & Křišťan, 2007*). Now we have turned to update summarization and thus we were able to participate for the first time in TAC/DUC evaluation series.

Our approach follows what has been called a term-based approach (*Hovy & Lin, 1997*). In term-based summarization, the most important information in documents is found by identifying their main terms, and then extracting from the documents the most important information about these terms. However, latent semantic analysis provides a way how to work with topics of the documents instead of terms only.

In this paper we first describe the classical LSA summarization model on which our previous methods were based (chapter 2). Moreover, we discuss here the Iterative Residual Rescaling (IRR) (Ando & Lee, 2001) modification that tries to fight with dominant topics. Chapter 3 covers our previous research. The single-document approach and its extension to process a set of documents is described. Chapter 4 contains the core of the paper. Our new sentence-extractive update summarizer is proposed. It uses the LSA representation modified by IRR. This representation uncovers topic/sentence distribution in the documents. Then it specifies a topic novelty value that combines topic significance within the summarized documents and topic redundancy measured on the basis of the reader's prior knowledge obtained from the set of older documents. And moreover, we have improved the sentence selection algorithm in order to prevent inner summary similarity. Prior to producing and sending out our summaries of TAC data we experimented with the DUC'07 corpus in order to set the summarizer's parameters and to compare to those systems that participated in the update summarization pilot task last year (chapter 5). Also results on TAC data are

briefly discussed there. Finally, in the last chapter, we conclude the paper and reveal our next point of focus in summarization research.

2 Latent Semantic Analysis Model for Summarization

Latent semantic analysis[1] is a fully automatic mathematical/statistical technique which is able to extract and represent the meaning of words on the basis of their contextual usage. Its fundamental idea is based on the fact that mutual similarity among the meanings of words or phrases can be obtained from the accumulated contexts in which the word or the phrase occurs and in which it does not. LSA was applied to various tasks: e.g. information retrieval (*Berry et al., 1995*), text segmentation (*Choi et al., 2001*), or document categorization (*Lee et al., 2006*). The first LSA application in text summarization was published in the year 2002 (*Gong & Liu, 2002*).

2.1 The Classical LSA Model

The heart of LSA-based summarization is a document representation[2] developed in two steps. In the first step we construct the terms[3] by sentences association matrix *A*. Each element of *A* indicates the weighted frequency of a given term in a given sentence. Having *m* distinguished terms and *n* sentences in the document(s) under consideration the size of *A* is *m* x *n*. Element a_{ij} of *A* represents the weighted frequency of term *i* in sentence *j* and is defined as:

$$a_{ij} = L(i, j) \cdot G(i), \tag{1}$$

where *L(i, j)* is the local weight of term *i* in sentence *j* and *G(i)* is the global weight of term *i* in the document. The weighting scheme we found to work best uses a binary local weight and an entropy-based global weight:

$$L(i, j) = 1 \text{ if term } i \text{ appears at least once in sentence } j, \text{ otherwise } L(i, j) = 0 \tag{2}$$

$$G(i) = 1 - \sum_j \frac{p_{ij} \log p_{ij}}{\log n}, \; p_{ij} = \frac{t_{ij}}{g_i}, \tag{3}$$

where t_{ij} is the frequency of term *i* in sentence *j*, g_i is the total number of times that term *i* occurs in the whole document and *n* is the number of sentences in the document.

The next step is to apply the Singular Value Decomposition (SVD) to matrix *A*. The SVD of an *m* x *n* matrix is defined as:

$$A = USV^T, \tag{4}$$

where *U* (*m* x *n*) is a column-orthonormal matrix, whose columns are called left singular vectors. The matrix contains representations of terms expressed in the newly created (latent) dimensions. *S* (*n* x *n*) is a diagonal matrix, whose diagonal elements are non-negative singular values sorted in descending order. V^T (*n* x *n*) is a row-orthonormal matrix which contains representations of sentences expressed in the latent

[1] The title used in information retrieval terminology is "latent semantic indexing". Basically, the terms are indexed into the latent space in which words with similar meaning are closer to each other than in the original one.
[2] In the case of multi-document summarization it is the representation of all documents assigned to the topic.
[3] We have used only words for now.

dimensions. The dimensionality of the matrices is reduced to *r* most important dimensions and thus, we receive matrices *U'* (*m* x *r*), *S'* (*r* x *r*) a V'^{T} (*r* x *n*). The optimal value of *r* can be learned from the training data.

From the mathematical point of view SVD maps the m-dimensional space specified by matrix *A* to the *r*-dimensional singular space. From an NLP perspective, what SVD does is to derive the latent semantic structure of the document represented by matrix *A*: i.e. a breakdown of the original document into r linearly-independent base vectors which express the main 'topics' of the document. SVD can capture interrelationships among terms, so that terms and sentences can be clustered on a 'semantic' basis rather than on the basis of words only. Furthermore, as demonstrated in (*Berry et al., 1995*), if a word combination pattern is salient and recurring in a document, this pattern will be captured and represented by one of the singular vectors. The magnitude of the corresponding singular value indicates the importance degree of this pattern within the document. Any sentences containing this word combination pattern will be projected along this singular vector, and the sentence that best represents this pattern will have the largest index value with this vector. Assuming that each particular word combination pattern describes a certain topic in the document, each singular vector can be viewed as representing such a topic (*Ding, 2005*), the magnitude of its singular value representing the degree of importance of this topic.

2.2 Iterative Residual Rescaling (IRR)

In *Ando & Lee (2001)* the topic dominancy problem of LSA representation was discussed. They showed that when the topic-sentence distribution is non-uniform in the analyzed text[4], the dominant topics take more than one dimension in the latent space, although the dimensions are orthogonal. If we consider a dominant topic, its first dimension is correct. However, the next dimensions do not correspond to the next topics but only to the residuals of the first dimension. Including the residuals though spoils the topic/sentence representation. Minor topics need not in this case be reflected in the representation at all after the dimensionality reduction cut. In order to resolve this problem, the Iterative Residual Rescaling (IRR) algorithm was proposed. This algorithm modifies the computation of singular decomposition. By amplifying the length differences among residual vectors (changing their scale) IRR boosts the influence of minority-topic sentences. The following figures 1-3 illustrate the whole process.

Figure 1. The first singular vector points in the dominant direction.

Figure 1 demonstrates two topics contained in the text. The topic on the right is the dominant one, a large number of sentences deals with it. The second topic is less important; only two sentence vectors represent it in the figure. The first singular vector u_1 points in the direction of the dominant topic. Figure 2 shows how the cumulative influence of a large number of small residuals for a major topic can cause smaller topics to be ignored; u_2 is still biased towards the dominant topic. Figure 3 illustrates how the influence of the minor topic is increased when the residual vectors are rescaled.

[4] Newspaper texts used for summarization, like those for the TAC evaluation, usually contain a dominant topic (a dominant linear combination of terms).

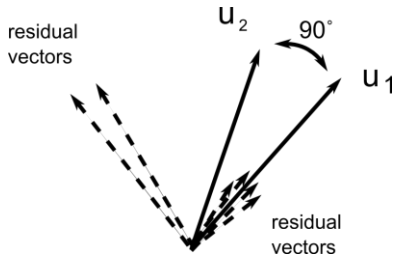

Figure 2. The second singular vector is still biased towards dominant-topic vectors, despite being orthogonal to the first one.

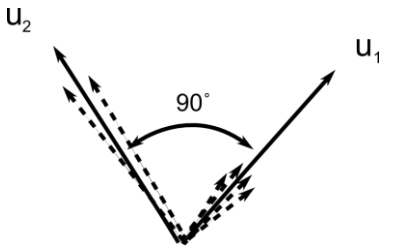

Figure 3. Rescaling the residuals boosts the influence of the minority-topic sentences.

The high-level pseudo-code for the IRR algorithm:

$R = A$ /* use matrix A to initiate the residual vectors */
for j from 1 to r /* create r dimensions */
$R_s = [| r_1 |^q r_1, \ldots, | r_n |^q r_n]$ /* rescale residuals */
$u_j =$ first singular vector of the SVD of matrix R_s
$R = R - u_j u_j^T R$ /* "subtract" the information contained in the created j-th topic from the residual vectors */
$A' = U^T A$ /* new representation of sentences (columns of A') */
$S V^T = A'$ /* decompose matrix A' into diagonal S and V^T in order to get the same representation as in the case of the classical LSA model */

We determine singular values in matrix S as lengths of row vectors in matrix A'. Matrix V^T is created from matrix A' by normalizing its rows. Scaling factor q controls the scale of long and short residuals. SVD is a special case in which q=0. This factor can be automatically determined by the AUTO-SCALE method – for details see *Ando & Lee (2001)*[5]:

$$q = \alpha \left(\frac{\left\| A^T A \right\|_F}{n} \right)^2 + \beta , \tag{5}$$

where coefficients α and β were empirically set in *Ando & Lee (2001)*: $\alpha = 3.5$ a $\beta = 0$.

The second parameter of the algorithm – r – is the number of desired dimensions. One way of setting the parameter is to train it on some training data. *Ando & Lee (2001)* found that learning thresholds on the basis of the residual ratio $\|R^{(j)}\|_F^2/n$ as a stopping criterion is effective. Intuitively, this ratio describes how much is left out of the proposed subspace. We do not want to reproduce the input matrix exactly, hence the threshold. In section 5.1 we describe our approach to training the threshold.

[5]The Frobenius norm $\|X\|_F$ is defined as $\sqrt{(\Sigma_{i,j} X[i,j]^2)}$.

To conclude, the IRR modification refines the topic-sentence representation. The dominant topics do not take more than one dimension and minor topics have a chance of being included in the representation.

3 Summarization Based on Latent Semantic Analysis

The work of *Gong & Liu (2002)* was the starting point for our own work. Our experiments with LSA-based summarization resulted in the modified single-document summarization method (*Steinberger & Ježek, 2004*), which we later extended to work in multi-document summarization (*Steinberger & Křišťan, 2007*). Some other summarization approaches, which more or less use the classical LSA model, appeared in the meantime (*Murray et al., 2005*; *Hachey et al., 2005*; *Yeh et al., 2005*).

3.1 Single-Document LSA-Based Summarization

The summarization method proposed by *Gong and Liu (2002)* uses the representation of a document described in section 2.1 to choose the sentences to go in the summary on the basis of the relative importance of the topics they mention, described by the matrix V^T. The summarization algorithm simply chooses for each topic the most important sentence for that topic: i.e., the *k*th sentence chosen is the one with the largest index value in the *k*th right singular vector in matrix V^T.

The main drawback of Gong and Liu's method is that when *l* sentences are extracted the top *l* topics are treated as equally important. However, in most cases the document contains one dominant topic which should dominate in the summary as well. Therefore, we proposed in (*Steinberger & Ježek, 2004*) the following modification: In matrix $B = S' \cdot V'^T$ the topic importance will be respected because the topic vectors (right singular vectors) will be scaled by the corresponding singular values that carry the topic importance. We changed the selection criterion to include in the summary sentences whose vectorial representation in matrix *B* has the greatest length, instead of sentences containing the highest index value for each topic. Intuitively, the idea is to choose the sentences with greatest combined weight across all topics, possibly including more than one sentence about an important topic, rather than always choosing one sentence for each topic as done by *Gong & Liu (2002)*. More formally, we measure the length s_j of each sentence vector in *B*:

$$s_j = \sqrt{\sum_{i=1}^{r} b_{ij}^2} \,, \tag{6}$$

where s_j is the length of the vector of *j*th sentence in the modified latent vector space, and its significance score for summarization too. We then include in the summary the sentences with the highest values in vector *s*. We showed (*Steinberger & Ježek, 2004*) that this modification results in a significant improvement over Gong and Liu's method.

3.2 Multi-Document LSA-Based Summarization

In *Steinberger & Křišťan (2007*) we modified the method described in the previous section to work in multi-document summarization. The input is a set of documents $C = \{D_1, D_2, \ldots, D_d\}$ related to a topic defined by the title and narrative. The columns of matrix *A*, which is then passed to SVD, are covered by weighted term vectors of all sentences in documents set *C*. The weighting scheme stays the same as described in 2.1, however, the global weight is computed separately for each document and the values that correspond to the terms contained in the topic narrative are multiplied by a coefficient that can adjust their greater importance. Singular value decomposition is then applied to matrix *A*. The score for each sentence

is computed in the same way as in the case of single-document summarization and the sentences with the highest score are selected for the summary.

4 Update Summarization Based on the LSA/IRR Model

In update summarization, we assume the reader's prior knowledge of the topic. The input consists of a set of older documents C_1, which represents the prior knowledge, and a set of newer documents C_2, which is intended for own summarization. The first step is to obtain a set of topics of the prior knowledge (denoted as "old" topics) and the set of new topics. Thus we perform the analysis of sets C_1 a C_2 separately: an input matrix is created for each set – A_1, respectively A_2. Experiments show that the best weighting system is again the Boolean local weight and the entropy-based global weight computed for each document. The values that correspond to terms in the narrative do not get any advantage at this stage because sentence vectors must be normalized in order to be used as an input to IRR. As one of the results of applying the IRR to the input matrices, we get matrices U_1 and U_2, whose columns contain topics of the analyzed sets of documents expressed in linear combinations of original terms. For each "new" topic t (a column of U_2) the most similar "old" topic is found (a column of matrix U_1). The similarity value indicates the redundancy of topic t – red_t:

$$red_t = \max_{i=1}^{r_1} \frac{\sum_{j=1}^{m} U_2[j,t] \cdot U_1[j,i]}{\sqrt{\sum_{j=1}^{m} U_2[j,t]^2} \cdot \sqrt{\sum_{j=1}^{m} U_1[j,i]^2}}, \tag{7}$$

where r_1 is the number of old topics (the number of latent dimensions arising from the decomposition of A_1). Thus the topic redundancy will be large if a similar topic is found in the set of older documents.

The topic importance is represented by its corresponding singular value (s_t). For each topic t we can compute topic novelty nov_t:

$$nov_t = (1 - red_t) \cdot s_t . \tag{8}$$

From topic novelties we create diagonal matrix NOV, in which the diagonal consists of nov_1, nov_2, …, nov_{r2}. Final matrix F can then be computed as $F = NOV \cdot V^T$. In this matrix, both the importance and novelty of the new topics are taken into account.

Sentence selection starts with the sentence that has the longest vector[6] in matrix F (the vector, the column of F, is denoted as f_{best}). After placing it in the summary, the topic/sentence distribution is changed by subtracting the information contained in that sentence:

$$F = F - \frac{f_{best} \cdot f_{best}^T}{\left| f_{best} \right|^2} \cdot F , \tag{9}$$

[6] We experimented with boosting the score of sentences that contain narrative terms. A slight improvement, but not statistically significant, was observed.

The vector lengths of similar sentences are decreased, thus preventing inner summary redundancy. After the subtraction the process of selecting the sentence that has the longest vector in matrix *F* and subtracting its information from *F* is iteratively repeated until the required summary length is reached.

5 Experiments

The aim of the first experiments with the proposed update summarizer was to find the optimal parameter values – the level of dimensionality reduction and scaling factor for IRR. For this training we could use the data from the DUC'07 pilot task. Moreover, we could see the comparison of our system with those that participated in the pilot last year. Then we generated update summaries for TAC texts. The results are presented in section 5.2.

5.1 Results of DUC'07 Data

The DUC 2007 corpus contains 10 topics. In each of them there are three sets of documents (A, B, and C). Each set contains up to 10 documents. In set A there are the oldest documents, in set B there are newer documents and in C there are the newest documents. The task is to create a summary for each set under the assumption that the reader has already read the older set(s) of documents. When summarizing set A, we cannot use any prior knowledge of the topic and thus it is a simple multi-document summary. In this case, the redundancy of each topic is set to zero in our algorithm.

Firstly, we needed to set the threshold for dimensionality reduction. The threshold *dmk* is defined by the following stopping criterion in the IRR iterative computation:

$$dmk < \frac{\left\|R^{(j)}\right\|_F^2}{\left\|A\right\|_F^2}. \tag{10}$$

If the criterion is true, thus the threshold is larger than the ratio of squares of Frobenius norms of the matrix of residual vectors in *j*th iteration $R^{(j)}$ and the initial matrix A ($A = R^{(0)}$), then the computation continues, otherwise the computation is finished. In other words, the computation finishes when the residual of matrix A drops below a percentage of the initial matrix. In our first run (TAC run 25), the *dmk* was set to 0.8. In the second priority run (TAC run 51), the threshold was set to 0.9, but in the denominator in formula 10 $||A||$ was substituted by $||R^{(1)}||$. So in this case the computation finished when the information in the residual matrix measured by the Frobenius norm dropped below 90% of the information in the residual matrix after the first iteration (not the initial matrix). Simply put, in the first run only a small number of dimensions (topics) appeared in the latent representation and in the second run there were more of them.

The optimal setting for the scaling factor for both runs was $\alpha = 0$ and $\beta = 3$.

Results with different thresholds were evaluated by ROUGE. In the first run, the ROUGE-2 score was maximized and in the second run ROUGE-SU4 was maximized.

Compared to the summarizers that participated in DUC'07 (24 summarizers in total), the first run was ranked 4th in ROUGE-2, when only one system was statistically significantly better, and 9th in ROUGE-SU4, when again only one system was significantly better. The second run was ranked 7th in ROUGE-2, when again just one system performed significantly better, and 8th in ROUGE-SU4; only the best system outperformed it significantly.

5.2 Results in TAC 2008

This year's TAC corpus contained 48 topics and two sets of documents for each – A (older documents) and B (newer documents). Both sets contained 10 documents. The target summary length was again 100 words. Only the summary for set B could use the prior knowledge from older documents in A and thus it was the true update summary. The summary for set A was a simple multi-document summary. In total, 71 summarizers[7] participated in the large-scale evaluation. The main evaluation approach was the Pyramid method (*Nenkova & Passonneau, 2005*). The results of our summarizer were promising – Table 1. In all major metrics, except for the average number of repetitions, our first run was ranked among the top 20%. The second run seems to be worse, which assumes that the larger dimensionality reduction cut works better for such short summaries. An interesting point is that although we performed just simple sentence extraction without any sentence modifications, the linguistic quality was pretty good compared to the other systems. However, these numbers mix the update summaries and the simple multi-document summaries. Thus, the influence of our innovative part of the summarizer cannot be clearly seen. Thanks to Guy Lapalme's Excel sheets, we can look at the results when just update summaries are used for the evaluation – Table 2. Even better results suggest a good performance of the summarizer's update component.

Table 1. Overall TAC results of our summarizer.

Evaluation metric	Rank of run 25 (Total No. of runs)	Rank of run 51 (Total No. of runs)
Average modified (pyramid) score	10 (58)	16 (58)
Average num. of SCUs	12 (58)	17 (58)
Average num. of repetitions	55 (58)	22 (58)
Macroavg. modified score with 3 models	10 (58)	16 (58)
Average linguistic quality	10 (58)	8 (58)
Average overall responsiveness	9 (58)	14 (58)
ROUGE-2	17 (71)	22 (71)
ROUGE-SU4	17 (71)	18 (71)
BE	13 (71)	15 (71)

Table 2. Separate TAC results of our update summaries.

Evaluation metric	Rank of run 25 (Total No. of runs)	Rank of run 51 (Total No. of runs)
Average modified (pyramid) score	7 (58)	12 (58)
Average num. of SCUs	9 (58)	12 (58)
Average linguistic quality	5 (58)	12 (58)
Average overall responsiveness	15 (58)	16 (58)
ROUGE-2	18 (71)	25 (71)
ROUGE-SU4	17 (71)	21 (71)
BE	13 (71)	25 (71)

[7] More precisely summarizer runs because each group could submit up to 3 runs.

6 Conclusion

Update summarization introduces another feature to summarization – the assumption of a prior knowledge of the summarized topic. Our method tries to determine the topics of the summarized set of documents and express in numbers their novelty. We described the IRR modification of the basic LSA summarization model that makes the LSA representation of topic/sentence distribution more reliable. The advantage of the method is that it works just with the context of terms and thus it is completely language independent. The summarizer was trained using DUC'07 data and 100-word summaries. However, we do not have data to see what needs to be changed when producing longer summaries. We participated in the TAC (DUC) evaluation campaign for the first time and the results seem to be very good. For next year, we plan to focus on better sentence ordering in the summary, clause-level sentence compression, and using co-reference relations.

This research was partly supported by project 2C06009 (COT-SEWing).

References

Ando R.K. & Lee L. (2001). Iterative Residual Rescaling: An Analysis and Generalization of LSI. In *Proceeding of the 24th SIGIR*.

Berry M.W. & Dumais S.T. & O'Brien G.W. (1995). Using linear algebra for intelligent IR. In *SIAM Review 37(4)*.

Choi F.Y.Y. & Wiemer-Hastings P. & Moore J.D. (2001). Latent Semantic Analysis for Text Segmentation. In *Proceedings of EMNLP*.

Ding Ch. (2005). A probabilistic model for latent semantic indexing. In *Journal of the American Society for Information Science and Technology 56(6)*.

Gong Y. & Liu X. (2002). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of ACM SIGIR*.

Hachey B. & Murray G. & Reitter D. (2005). The EMBRA system at DUC 2005: Query-oriented multi-document summarization with a very large latent semantic space. In *Proceedings of the Document Understanding Conference*.

Hovy E. & Lin C. (1997). Automated text summarization in summarist. In *ACL/EACL workshop on intelligent scalable text summarization*, Madrid, Spain.

Landauer T.K. & Dumais S.T. (1997). A solution to platos problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. In *Psychological Review 104*.

Lee C.-H. & Yang H.-C. & Ma S.-M. (2006). A Novel Multilingual Text Categorization System using Latent Semantic Indexing. In *Proceedings of the First International Conference on Innovative Computing, Information and Control*, IEEE Computer Society.

Murray G. & Renals S. & Carletta J. (2005). Extractive Summarization of Meeting Recordings. In *Proceedings of Interspeech*.

Nenkova A. & Passonneau R. (2005). Evaluating Content Selection in Summarization: The Pyramid Method. In *Document Understanding Conference*.

Steinberger J. & Ježek K. (2004). Text Summarization and Singular Value Decomposition. In *Lecture Notes for Computer Science 2457*, Springer-Verlag.

Steinberger J. & Křišťan M. (2007). LSA-Based Multi-Document Summarization. In *Proceedings of 8th International PhD Workshop on Systems and Control*.

Yeh J.Y. & Ke H.R. & Yang W.P & Meng I.H (2005). Text summarization using a trainable summarizer and latent semantic analysis. In *Special issue of Information Processing and Management on An Asian digital libraries perspective 41(1).*