ISSN 0361-7688, Programming and Computer Software, 2011, Vol. 37, No. 6, pp. 299–305. © Pleiades Publishing, Ltd., 2011.
Original Russian Text © I.V. Mashechkin, M.I. Petrovskiy, D.S. Popov, D.V. Tsarev, 2011, published in Programmirovanie, 2011, Vol. 37, No. 6.

Automatic Text Summarization Using Latent Semantic Analysis

I. V. Mashechkin, M. I. Petrovskiy, D. S. Popov, and D. V. Tsarev

Department of Computational Mathematics and Cybernetics, Moscow State University, Moscow, 119991 Russia
e-mail: mash@cs.msu.su, michael@cs.msu.su, ixaphire@gmail.com, dima.tsarev@gmail.com

Received May 24, 2011

Abstract—In the paper, the most state-of-the-art methods of automatic text summarization, which build summaries in the form of generic extracts, are considered. The original text is represented in the form of a numerical matrix. Matrix columns correspond to text sentences, and each sentence is represented in the form of a vector in the term space. Further, latent semantic analysis is applied to the matrix obtained to construct sentences representation in the topic space. The dimensionality of the topic space is much less than the dimensionality of the initial term space. The choice of the most important sentences is carried out on the basis of sentences representation in the topic space. The number of important sentences is defined by the length of the demanded summary. This paper also presents a new generic text summarization method that uses nonnegative matrix factorization to estimate sentence relevance. Proposed sentence relevance estimation is based on normalization of topic space and further weighting of each topic using sentences representation in topic space. The proposed method shows better summarization quality and performance than state-of-the-art methods on the DUC 2001 and DUC 2002 standard data sets.

DOI: 10.1134/S0361768811060041

1. INTRODUCTION

The paper is devoted to one of the most popular problems of text mining, namely, automatic text summarization [1, 2]. This is one of the basic algorithmic problems arising in the implementation of many application systems, such as data mining of text databases (e.g., Oracle Text), filters for web-based information retrieval (as early as 1998, Inxight Summarizer was used for constructing summaries in AltaVista), and word processing tools (e.g., AutoSummarize in Microsoft Office).

Methods of automatic text summarization considered in this paper build summaries in the form of generic extracts; i.e., the resulting document summary is a sequence of fragments of the original text. A sentence is usually used to express context in summarization. However, for instance, a context can be represented by a paragraph for longer documents. Further, we use sentences as fragments. Moreover, we assume that summaries are constructed for a broad community of readers; i.e., all main topics of the original text are to be presented rather than only certain topics of interest for particular readers.

The most popular methods of automatic summarization that construct summaries of the above-described class are based on the use of latent semantic analysis (LSA). In these methods, the original text is represented in the form of a numerical matrix. Matrix columns correspond to text sentences (or other fragments), and each sentence is represented in the form of a vector in the term space. Further, LSA is applied to the matrix obtained to construct sentences representation in the topic space. The dimensionality of the topic space is much less than the dimensionality of the initial term space. The choice of the most important sentences is carried out on the basis of sentences representation in the topic space. The number of important sentences is defined by the length of the demanded summary.

The paper is organized as follows. In Section 2, a survey of the existing formal models of text representation in the matrix form is given. In Section 3, automatic summarization methods based on latent semantic analysis are described. To construct representations of text sentences in the topic space, LSA uses matrix factorizations, such as singular value decomposition (SVD) and nonnegative matrix factorization (NMF). In addition, in this section, we present our own method of automatic text summarization that is based on NMF of the original text matrix. Section 4 is devoted to experimental study of the considered methods on the DUC 2001 and DUC 2002 standard data sets.

2. MODELS OF DATA TEXT REPRESENTATION

Latent semantic analysis works with a collection of all text sentences. The original text is represented in the form of a numerical matrix. Matrix columns correspond to text sentences, and each sentence is represented as a vector a of fixed dimension n, where n is the

$$\begin{bmatrix} \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ \dots \\ a_{m,1} \end{bmatrix} \begin{bmatrix} a_{1,2} \\ a_{2,2} \\ \dots \\ a_{m,2} \end{bmatrix} \dots \begin{bmatrix} a_{1,n} \\ a_{2,n} \\ \dots \\ a_{m,n} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} u_{1,1} \\ \dots \\ u_{m,1} \end{bmatrix} \dots \begin{bmatrix} u_{1,k} \\ \dots \\ u_{m,k} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \sigma_1 & \dots & 0 \\ & \dots & \\ 0 & \dots & \sigma_k \end{bmatrix} \begin{bmatrix} \begin{bmatrix} v_{1,1} \\ \dots \\ v_{1,k} \end{bmatrix} \dots \begin{bmatrix} v_{n,1} \\ \dots \\ v_{n,k} \end{bmatrix} \end{bmatrix}$$

A_k U_k Σ_k V_k^T

Fig. 1. SVD approximation of matrix A.

number of text features and the ith element of the vector determines weight of the ith feature. To define a representation model, it is required to define a feature space and select a method of weight calculation.

Vector space model (or term vector model) is the most frequently used algebraic model for text representation as vectors of identifiers, such as, for example, index terms. In this method, each feature corresponds to a separate term. Terms are generally meant to be various words of the text. Usually, certain preliminary text processing is applied to the text with the aim to obtain more "informative" feature space.

The purpose of the preliminary text processing is to leave only most informative features that characterize the summarized text most of all. Moreover, the reduction of the number of features to be analyzed results in the reduction of the required computational resources.

The standard test data sets of documents DUC 2001 and DUC 2002 used in this work are texts in English. Therefore, preliminary text processing for formation of the list of terms included methods of stop-words filtration and stemming [3].

The original text is represented as a term-by-sentences matrix $A = [A_1, A_2, ..., A_n]$, the rows of which correspond to the text terms and columns, to its sentences. Each sentence of the original text is represented as a vector in the term space, the coordinates of which are weight coefficients of the corresponding terms. Formally, the jth text sentence is mapped to vector $A_j = [a_{1,j}, a_{2,j}, ..., a_{m,j}]^T$, where m is the number of text terms, $a_{i,j} = L_{i,j}G_i$, $L_{i,j}$ is a local weight of term i in sentence j, G_i is a global weight of term i in the entire original text. Below, we present the most popular local and global term weights, which are used in various problems of text mining [4, 5].

Local weights:

• Frequency weight (*FQ*, *TF*), the number of occurrences of term i in sentence j, $L_{i,j} = t_{i,j}$.

• Binary weight (*BI*) $L_{i,j} = \chi(t_{i,j}) = \begin{cases} 1, & \text{if } t_{i,j} > 0 \\ 0, & \text{if } t_{i,j} = 0. \end{cases}$

• Logarithmic weight (*LOG*) $L_{i,j} = \log(1 + t_{i,j})$.

• Normalized logarithmic weight (*LOGN*) $L_{i,j} = \begin{cases} \dfrac{1 + \log(t_{i,j})}{1 + \log(a_j)}, & \text{if } t_{i,j} > 0 \\ 0, & \text{if } t_{i,j} = 0, \end{cases}$ where a_j is an average number of term occurrences in sentence j.

• Augmented weight (*AU*) $L_{i,j} = 0.5\chi(t_{i,j}) + 0.5\left(\dfrac{t_{i,j}}{x_j}\right)$, where $x_j = \max\limits_i(t_{i,j})$.

Global weights:

• No weight (*NW*) $G_i = 1$.

• Inverse document frequency (*IDF*) $G_i = \log\left(\dfrac{N}{n_i}\right) + 1$, where N is the total number of all sentences in the text and n_i is the number of sentences containing term i.

• Entropy (*EN*) $G_i = 1 - \sum_{j=1}^{N}\left(\dfrac{p_{i,j}\log p_{i,j}}{\log N}\right)$, where $p_{i,j} = \dfrac{t_{i,j}}{F_i}$ and $F_i = \sum_{k=1}^{N} t_{i,k}$.

• Global frequency (*GF*) $G_i = \dfrac{F_i}{n_i}$.

3. AUTOMATIC SUMMARIZATION METHODS

As noted above, the original text is represented as term-by-sentences matrix. Matrix columns correspond to text sentences, and each sentence is represented as a vector in the term space. Further, latent semantic analysis is applied to the matrix obtained to construct text sentences representation in the topic space. The dimensionality of the topic space is much less than the dimensionality of the original term space. Based on the obtained representations, the most significant text sentences are selected. The number of these sentences depends on the length of the demanded summary.

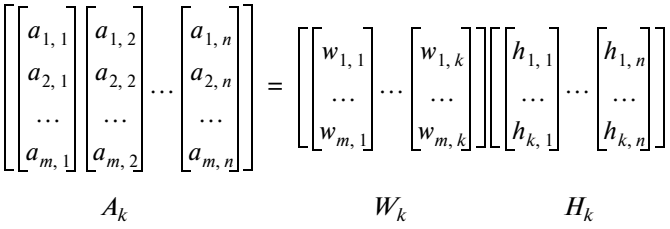

Fig. 2. NMF approximation of matrix A.

3.1 Latent Semantic Analysis

Latent semantic analysis (LSA) is a completely automated algebraic–statistical method of processing text information presented in a natural language, which is used for obtaining and representing context use of word meanings in text sentences (or in a set of text documents). The basic idea of this method is that the collection of all sentences of the original text leads to mutual restrictions on word usage, and these restrictions determine similarity of semantic meanings of words and text sentences [2, 6, 7]. Latent semantic analysis is widely used in various fields of text mining, including information retrieval[1] [7], document categorization, automatic summarization [2, 8], and so on. In what follows, LSA will be considered in the context of the automatic summarization problem.

LSA works with a term-by-sentences matrix representation of the text. Thus, the original text is represented as a matrix $A \in \Re^{m \times n}$, where m is the number of different terms and n is the number of sentences in the text. The next step of LSA is construction of representations of text sentences in the topic space, which is performed by applying one of matrix factorizations to the text matrix A. The first, and so far most popular, factorization is singular value decomposition (SVD) [6–8]. In this paper, we also consider nonnegative matrix factorization (NMF) [9–12].

3.2 Singular Value Decomposition

Suppose that the original text is represented as a term-by-sentences matrix $A \in \Re^{m \times n}$, where m is the number of different terms and n is the number of selected sentences in the text. Usually, the number of different terms is greater than the number of sentences; therefore, without loss of generality, we may assume that $m \geq n$.

Then, singular value decomposition of matrix $A \in \Re^{m \times n}$ is defined as [13]

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T,$$

where $U \in \Re^{m \times n}$ is a matrix composed of orthonormal columns, which are called left singular vectors (u_i); $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n) \in \Re^{m \times n}$ is a diagonal matrix such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0 = \sigma_{r+1} = \ldots = \sigma_n$, $r = \mathrm{rank}(A) \leq \min(m, n)$ is the rank of matrix A, and σ_i are singular values of matrix A; and $V \in \Re^{m \times n}$ is an orthonormal matrix whose columns are called right singular vectors (v_i).

Singular value decomposition is widely used in various problems of text mining, since it gives us the best approximation of the original matrix A by matrix $A_k = U_k \Sigma_k V_k^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T$ of rank $k \ll \min(m, n)$ in the Frobenius norm ($\|A\|_F = \|A\|_2 = \sqrt{\sum_{i,j} |a_{ij}|^2}$) [13].

Each column i of matrix A corresponding to the vector of sentence i in text is mapped to column i of matrix V_k^T, which represents sentence i in the space of k topics. Matrix U_k specifies mapping between the space of k topics and the space of m terms [8]. Singular values σ_l, where $1 \leq l \leq k$, denote weights of selected text topics (Fig. 1).

Computational complexity of singular value decomposition of matrix $A \in \Re^{m \times n}$ is $O(\min(nm^2, mn^2)) = O(mn^2)$. However, for sparse matrices, complexity of SVD is $O(mnc)$, where c is an average number of nonzero elements in the matrix columns, i.e., an average number of different terms in the text sentences.

3.3 Nonnegative Matrix Factorization

Nonnegative matrix factorization is frequently used in LSA. In particular, it is used in cluster analysis and in topic extraction tasks [9–12].

We have a term-by-sentences matrix of the text $A \in \Re^{m \times n}$, where m is the number of different terms and n is the number of sentences in the text. Elements of matrix A take nonnegative values, since they are weights of the corresponding terms in the sentences. Then, the purpose of NMF is to find nonnegative matrices $W_k \in \Re^{m \times k}$ and $H_k \in \Re^{k \times n}$ with nonnegative elements that minimize the functional $f(W_k, H_k) =$

[1] In information retrieval, this method is referred to as latent semantic indexing (LSI).

Table 1. Standard DUC data sets

Name of data set	The number of documents	The number of model summaries
DUC 2001	297	925
DUC 2002	533	1112

$\frac{1}{2}\|A - W_k H_k\|_F^2$, $k \ll \min(m, n)$. As a result, we obtain an approximation of matrix A: $A_k = W_k H_k$.

Matrix W_k specifies mapping between the space of k topics and the space of m terms, and matrix H_k corresponds to the representation of the sentences in the topic space. Since elements of matrix W_k are nonnegative, we can find out which terms of the text better characterize each of the topics, which are associated with the columns of matrix W_k. Thus, unlike SVD, nonnegative matrix factorization provides us with a well-interpreted semantic space (Fig. 2).

Basic differences of NMF from SVD are as follows:

• Elements of matrices W_k and H_k cannot take negative values.

• There is no requirement of orthogonality in the topic space; i.e., columns of matrix W_k are not necessarily orthogonal vectors.

• There is no matrix whose elements estimate weights of the selected topics.

In this work, NMF was performed by the most popular algorithm described in [9].

1. Elements of matrices $W_1 \in \Re_+^{m \times k}$ and $H_1 \in \Re_+^{k \times n}$ are taken to be random nonnegative values.

2. The following iteration formulas for calculation of matrices W and H are applied in a loop p times:

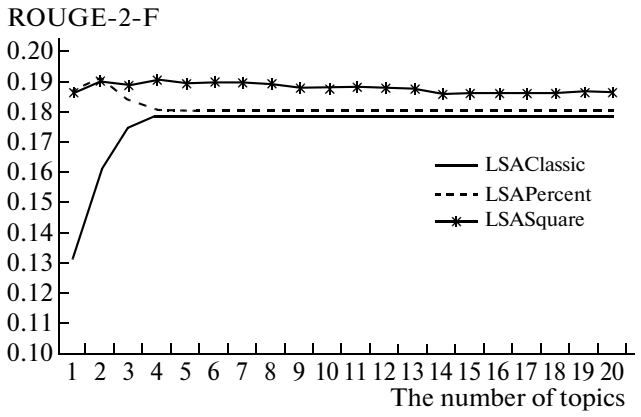

Fig. 3. LSAClassic, LSAPercent, and LSASquare summarization methods.

(a) $H_{b,j}^{p+1} = H_{b,j}^{p} \frac{((W^p)^T A)_{b,j}}{((W^p)^T W^p H^p)_{b,j}}$, $\forall b, j$: $1 \le b \le k$, $1 \le j \le n$,

(b) $W_{i,a}^{p+1} = W_{i,a}^{p} \frac{(A(H^{p+1})^T)_{i,a}}{(W^p H^{p+1}(H^{p+1})^T)_{i,a}}$, $\forall i, a$: $1 \le i \le m$, $1 \le a \le k$.

Computational complexity of this algorithm is $O(mnp)$.

3.4 Methods of Text Sentences Selection for a Summary

In the above, we considered methods for construction of sentences representation in the topic space. Below is a survey of methods for sentence relevance estimation, which are based on text sentences representation in the topic space. In addition, we present a new sentence relevance estimation, which is based on nonnegative matrix factorization of the original text matrix.

3.4.1 Methods based on singular value decomposition. When SVD is used, the approximation of the original text matrix can be written in the form $A_k = U_k \Sigma_k V_k^T$, $k \ll \min(m, n)$.

One of the following methods is used to select the most important text sentences:

• Topics are considered in the order from 1 through k. For a topic l, where $1 \le l \le k$, row $v_l^T = [v_{l,1}, v_{l,2}, ..., v_{l,n}]$ of matrix $V_k^T \in \Re^{k \times n}$ is considered. The elements of this row represent weights of topic l in n sentences. The element $v_{l,i}$ with the maximum absolute value is determined. This means that sentence i matches topic l better than others. If sentence i is already included in the summary, then the next in terms of absolute value element of vector v_l is selected, and so on (in what follows, we denote this method as LSAClassic) [8].

• The basic disadvantage of the above method is that sentences corresponding to a topic with small weight may occur in the summary. To get rid of this drawback, the number of sentences corresponding to the topics is selected based on the percent ratio of the topic weight to the sum of weights of all constructed topics (in what follows, we denote this method as LSAPercent) [8].

• The newest method is the method [2, 14] in which each sentence of the document is associated with a numeric value called sentence relevance. Then, a required number of sentences is selected in the decreasing order of their relevance measure. Sentence relevance i, where $1 \le i \le n$, is the length of column vector i of matrix $\Sigma_k^2 V_k^T$ (in what follows, we denote this method as LSASquare) [8].

3.4.2 A method based on nonnegative matrix factorization. When nonnegative matrix factorization is used, the approximation of the original text matrix $A \in \Re^{m \times n}$ can be written in the form $A_k = W_k H_k$, $k \ll \min(m, n)$. The newest and most popular method for selecting most important sentences on the basis of this representation is described in [15]. Its modifications are also used in query-based summarization [16] and in multi-document summarization [17].

The basic idea of this method consists in the calculation of generic relevance (GR) for sentences of the text. Generic relevance of the jth sentence is calculated by the formula $GR_j = \sum_{i=1}^{k} (h_{ij} weight(H_{i*}))$, where $weight(H_{i*}) = \dfrac{\sum_{q=1}^{n} h_{iq}}{\sum_{p=1}^{k}\sum_{q=1}^{n} h_{pq}}$ is relative relevance of the jth topic among all selected topics. Then, the required number of sentences with the greatest values of generic relevance is selected. This method is called generic relevance method.

3.4.3 The proposed approach based on nonnegative matrix factorization. As has already been noted, unlike in SVD, in NMF, there is no matrix of topic weights.

We propose a new method the basic idea of which consists in two-stage estimation of weights of the selected topics based on the obtained matrices $W_k = [w_{ij}]$ and $H_k = [h_{ij}]$.

On the first stage, the space of k topics is normalized; i.e., lengths of vector columns of matrix W_k are normalized to unity:

$$A_k = W_k H_k = NormW_k NormH_k,$$

where

$$NormW_k = W_k \mathrm{diag}(1/\|w^1\|, \ldots, 1/\|w^k\|),$$

$$NormH_k = \mathrm{diag}(\|w^1\|, \ldots, \|w^k\|) H_k,$$

$$\|w^l\| = \sqrt{\sum_{p=1}^{m} w_{pl}^2}, \quad 1 \le l \le k.$$

Columns of matrix $NormH_k = [normh_{ij}]$ correspond to n sentences in the normalized space of k topics. The kth row $NormH_k$ indicates weights of kth topic in all n sentences. The greater norm of rows of $NormH_k$, the greater weights of corresponding topics in all text. Based on this, the second stage of estimation of the topic weights consists in the calculation of the weight of topic l as the length of the lth row of matrix $NormH_k$:

$$\|normh_l\| = \sqrt{\sum_{q=1}^{n} normh_{lq}^2} = \|w^l\| \sqrt{\sum_{q=1}^{n} h_{lq}^2}$$

$$= \|w^l\| \|h_l\|, \quad 1 \le l \le k.$$

Thus, the weighted representation of n sentences in the normalized space of k topics is made to correspond to the matrix $WeightedH_k$:

$$WeightedH_k = \mathrm{diag}(\|w^1\| \|h_1\|, \ldots, \|w^k\|^2 \|h_k\|) H_k.$$

Then, relevance of each sentence is calculated as the sum of elements of the corresponding column vector in matrix $WeightedH_k$. Relevance of sentence j, where $1 \le j \le n$, is $R_j(WeightedH_k) = \sum_{i=1}^{k} (weightedh_{ij}) = \sum_{i=1}^{k} (\|w^i\| \|h_i\| h_{ij})$. This estimate of sentence relevance will be referred to as *extended relevance* (*ExtR*). Next, the required number of sentences with the greatest values of the extended relevance are selected. We will further refer to this method as the extended relevance method.

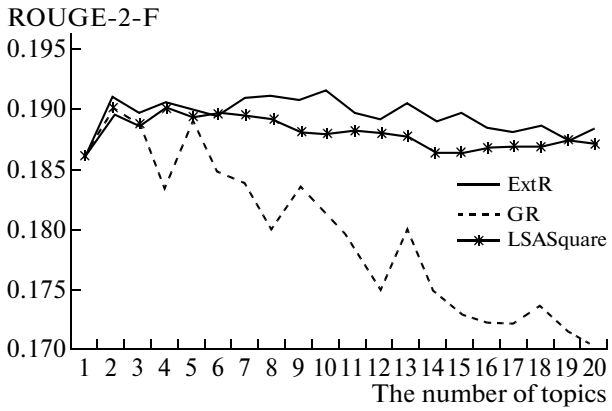

Fig. 4. Generic relevance (GR), extended relevance (ExtR), and LSASquare summarization methods.

4. RESULTS OF EXPERIMENTS

Currently, the most frequently used tool for evaluation of quality of summarization algorithms is the completely automated utility ROUGE (Recall-Oriented Understudy for Gisting Evaluation). It is sup-

Table 2. Comparison of the summarization methods

Measure (ROUGE)	SVD			NMF	
	LSAClassic	LSAPercent	LSASquare	GR	ExtR
ROUGE-2-F	0.17957	0.18082	0.18965	0.18023	0.19260
ROUGE-L-F	0.35415	0.35595	0.36816	0.35935	0.37229
ROUGE-S4-F	0.14054	0.14163	0.15146	0.14344	0.15390

plied with a set of summaries constructed by the algorithm and one (or more) model summary. Further, by means of special measures, the summaries are compared, and an estimate (average over all summaries) is assigned to the algorithm [18].

Standard data sets include text documents and model summaries to them, which are usually constructed by a human. Note that there may be several model summaries for one document, since, generally speaking, one summary may look nice for one person and bad for another. Data sets may differ depending on the kind of the summarization problem. To evaluate single-document summarization methods, DUC 2001 and DUC 2002 data sets may be used [19].

Analysis of applicability of different measures to various summarization problems is given in [18]. In particular, for evaluation of single-document algorithms on the DUC 2001 and DUC 2002 data sets, it is recommended to use measures ROUGE-2, ROUGE-L, ROUGE-S, and ROUGE-W. One of the results of work [18] is the conclusion that the DUC 2001 and DUC 2002 data sets possess a sufficient number of model summaries for adequate evaluation of quality of algorithms (Table 1).

In this work, quality of summarization algorithms was evaluated on the DUC 2001 and DUC 2002 data sets. To compare summaries, utility ROUGE with measures ROUGE-2, ROUGE-L, ROUGE-S, and ROUGE-W was used.

Before testing summarization methods, sentences of each document were mixed in a random order. This was done because the most informative sentences in documents from the DUC 2001 and DUC 2002 data sets are, as a rule, the first ones.

Evaluation of the above-described algorithms of automatic summarization on all DUC data sets used with various combinations of local and global weights showed that the best results were obtained when the local weight was binary and the global weight was entropy (weight scheme *BI*EN*).

Below are results of testing algorithms on the DUC 2002 data set, since this set has a greater number of documents and model summaries compared to DUC 2001. Moreover, results were similar on all DUC data sets.

The number of topics for LSA is selected much smaller than dimensionalities of the text matrix, i.e. $k \ll \min(m, n)$. For the DUC 2002 data set, the average number of document matrix rows is 239, and the average number of columns is 37. The plots presented in Figs. 3 and 4 show variation of the ROUGE-2 f-measure depending on the number of topics, where the number of topics varies from 1 to 20. ROUGE-2 f-measure has been selected because the results of various ROUGE measure usages are similar. Accordingly, the greater the value on the ROUGE-2-F axis, the higher the quality of summaries obtained by the algorithm.

Table 3. Run times of the LSASquare and extended relevance methods

Data set	LSASquare	ExtR
DUC 2002	18 min 40 s	16 min 30 s
DUC 2001	11 min 40 s	9 min 15 s

The plots presented in Fig. 3 show variation of quality of the summaries depending on the number of topics selected for the LSAClassic, LSAPercent, and LSASquare methods with SVD.

The experiments demonstrated that the best results were shown by the LSASquare method. It should also be noted that quality of summaries obtained by this method almost does not depend on the number of topics selected.

Results of testing generic relevance (GR) and extended relevance (ExtR) methods are presented in Fig. 4. This plot also shows results of the LSASquare method, which demonstrated the best results earlier.

The optimal number of topics (the number of semantic features) in latent semantic analysis is determined empirically for each problem [6]. In the summarization problem, the number of topics is usually determined depending on the size of the demanded summary [14].

Model summaries from the DUC 2001 and DUC 2002 sets consist, on the average, of 100 words. Therefore, quality of the automatic summarization algorithms is evaluated on these sets also with the help of summaries consisting of 100 words. Thus, for each document, by calculating the number of words in it, one can determine the percentage ratio $p\%$ of the summary to the document. If the size of the original text matrix is $m \times n$, then the number of topics is calculated as $k = \frac{p}{100} n$. Results of the experiments where the number of topics was calculated in this way are presented in Table 2 for different ROUGE measures.

Singular value decomposition was implemented with the help of MATLAB function svds(). Nonnegative matrix factorization was also implemented in MATLAB [20], with the number of iterations being limited to 300 (similar to the number of iterations in MATLAB function svds()).

Performance test was run on AMD Athlon(tm) Processor 3200+, RAM 2 Gb, HDD 200Gb, operating system Linux, kernel 2.6.34, distributive Arch Linux. The run times of the LSASquare and extended relevance methods are presented in Table 3.

The results presented in Figs. 3 and 4 and Tables 2 and 3 demonstrate that the proposed generic summarization method that uses extended relevance for the estimation of text sentences importance outperforms the state-of-the-art methods in terms of summarization quality and performance.

5. CONCLUSIONS

In the paper, state-of-the-art methods of automatic text summarization, which build summaries in the form of generic extracts, have been considered. The original text is represented in the form of a numerical matrix. Matrix columns correspond to text sentences, and each sentence is represented in the form of a vector in the term space. Further, latent semantic analysis is applied to the matrix obtained to construct sentences representation in the topics space. The dimensionality of the topic space is much less than the dimensionality of the initial term space. The choice of the most important sentences is carried out on the basis of sentences representation in the topic space. The number of important sentences is defined by the length of the demanded summary.

A new generic text summarization method that uses nonnegative matrix factorization to estimate sentence relevance has also been presented. The proposed sentence relevance estimation is based on normalization of topic space and further weighting of each topic using sentences representation in the topic space. The proposed method shows better summarization quality and performance than the considered state-of-the-art methods on the DUC 2001 and DUC 2002 standard data sets.

The results obtained make us conclude that the dimension reduction of the feature space (the space of the text terms) with the help of nonnegative matrix factorization and proposed estimation of semantic features (topics) weight better preserves internal structure of the text compared to the singular value decomposition.

REFERENCES

1. Mani, I. and Maybury, M.T., *Advance in Automatic Text Summarization*, Cambridge, Ma: The MIT Press, 1999.
2. Ježek, K. and Steinberger, J. Automatic Text Summarization (The State of the Art 2007 and New Challenges), *Proc. of Znalosti 2008*, Bratislava, 2008, pp. 1–12. http://textmining.zcu.cz/publications/Z08.pdf.
3. Garcia, E., Information Retrieval Tutorials: Document Indexing Tutorial. http://www.miislita.com/information-retrieval-tutorial/indexing.html.
4. Garcia, E., Vector Theory and Keyword Weights. http://www.miislita.com/term-vector/term-vector-1.html.
5. Chisholm, E. and Kolda, T.G., New Term Weighting Formulas for the Vector Space Method in Information Retrieval, *Tech. Rep. no. ORNL-TM-13756,* Oak Ridge National Laboratory, Oak Ridge, TN, March 1999.
6. Landauer, T.K. and Dumais, S.T., A solution to Plato's Problem: The Latent Semantic Analysis Theory of the Acquisition, Induction and Representation of Knowledge, *Psychological Rev.*, 1997, vol. 104, pp. 211-240.
7. Ye, Y., Comparing Matrix Methods in Text-based Information Retrieval, *Tech. Rep. School of Mathematical Sciences*, Peking University, 2000. http://dean.pku.edu.cn/bksky/2000jzlwj/39.pdf.
8. Gong, Y. and Liu, X., Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis, *SIGIR-2001*, 2001.
9. Lee, D.D. and Seung, H.S., Learning the Parts of Objects by Non-negative Matrix Factorization, *Nature*, 1999, vol. 401, pp. 788–791.
10. Wei Xu, Xin Liu, and Yihong Gong, Document Clustering Based on Non-negative Matrix Factorization, *Proc. of the 26th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Toronto, 2003.
11. Berry, M.W., Browne, M., Langville, A.N., Pauca, V.P., and Plemmons, R.J., Algorithms and Applications for Approximate Nonnegative Matrix Factorization, *Computational Statistics Data Analysis*, 2007, vol. 52, no. 1, pp. 155–173.
12. Rakesh, P., Shivapratap, G., Divya, G., and Soman, K.P., Evaluation of SVD and NMF Methods for Latent Semantic Analysis, *Int. J. Recent Trends Engineering*, 2009, vol. 1, no. 3.
13. Berry, M.W., Dumais, S.T., and O'Brien G.W., Using Linear Algebra for Intelligent Information Retrieval, *Univ. of Tennessee Knoxville*, TN, USA, 1994.
14. Steinberger, J., Text Summarization within the LSA Framework, *PhD Dissertation*, Univ. of West Bohemia in Pilsen, Czech Republic, 2007.
15. Ju-Hong Lee, Sun Park, Chan-Min Ahn, and Daeho Kim, Automatic Generic Document Summarization Based on Non-negative Matrix Factorization, *Information Processing Management: Int. J.*, 2009, pp. 20–34.
16. Sun Park, Personalized Summarization Agent Using Non-negative Matrix Factorization, *PRICAI 2008: Trends in Artificial Intelligence*, 2008.
17. Sun Park, Ju-Hong Lee, Deok-Hwan Kim, and Chan-Min Ahn, Multi-document Summarization Using Weighted Similarity between Topic and Clustering-based Non-negative Semantic Feature, in *Advances in Data and Web Management*, 2007
18. Lin, C.-Y., Looking for a Few Good Metrics: Automatic Summarization Evaluation - How many samples are enough?, *Proc. of NTCIR 2004*, Tokyo, 2004, pp. 1765–1776.
19. Document Understanding Conferences. http://duc.nist.gov.
20. DTU Toolbox. http://isp.imm.dtu.dk/toolbox/menu.html.