# Multi-document Summarization using Tensor Decomposition

Marina Litvak and Natalia Vanetik

Shamoon College of Engineering, Beer Sheva,
Israel

marinal@sce.ac.il, natalyav@sce.ac.il

**Abstract.** The problem of extractive text summarization for a collection of documents is defined as selecting a small subset of sentences so the contents and meaning of the original document set are preserved in the best possible way. In this paper we present a new model for the problem of extractive summarization, where we strive to obtain a summary that preserves the information coverage as much as possible, when compared to the original document set. We construct a new tensor-based representation that describes the given document set in terms of its topics. We then rank topics via Tensor Decomposition, and compile a summary from the sentences of the highest ranked topics.

**Keywords.** Tensor decomposition, multilingual multi-focument summarization.

## 1 Introduction

Automated text summarization is an active field of research in various communities such as Information Retrieval (IR), Natural Language Processing (NLP), and Text Mining (TM). Summarization is important for IR because it helps to access large repositories of textual data efficiently by identifying the essence of a document and indexing a repository. Taxonomically, we distinguish between *single-document* summarization, where a summary of a single document is generated, and *multi-document* summarization, where a summary of a cluster of related documents is generated. Also, we distinguish between an automatically generated *extract*—where the most salient fragments of the input documents (sentences, paragraphs, and so on) and an *abstract*—re-formulated synopsis expressing the main idea of the input documents. Because generating abstracts requires a deep linguistic analysis of the input documents, most existing summarizers work in extractive manner [22].

In this paper we deal with the problem of *multi-document extractive* summarization. In our approach, we strive to cover as many aspects (expressed by topics) of the summarized documents as possible. We generate topics by clustering similar sentences semantically, ranking those topics via Tensor Decomposition, and, finally, compiling a summary from representative sentences of the most ranked topics. Because the method includes only very basic linguistic analysis (see Section 3.2), which is optional, it can be applied to cross-lingual/multilingual summarization.

This paper is organized as follows: Section 2 depicts related work, Section 3 introduces problem setting and its solution, including a tensor representation model, a method for ranking topics, and the method used to compile the summary. Section 4 describes the experimental setup and results. The final section contains our future work and conclusions.

## 2 Related Work

Numerous techniques for automated summarization have been introduced in recent decades, trying to reduce the constant information overload of professionals in a variety of fields. Special attention has been focused on the problem of creating summaries from *multiple sources* in *multiple languages* [26]. This has partly been due to the rapid

ISSN 1405-5546
DOI: 10.13053/CyS-18-3-2026

growth of the amount of textual information in different languages freely accessible online. Systems that are able to summarize an enormous number of documents in the multilingual environment are usually based on mathematical or statistical models [12], similarity-based ranking [5], and other approaches that do not depend on a deep linguistic analysis and are able to represent high data volumes as a compact mathematical model.

Some authors reduce summarization to the maximum coverage problem [28]. This approach extracts sentences to a summary to cover as much information as possible, where information can be measured by text units such as terms or n-grams. Despite great performance [28] in the summarization field, the maximum coverage problem is known as NP-hard [18]. Some works attempt to find a near-optimum solution through a greedy approach [6, 28], while others try to find a more accurate approximated solution by a Integer Linear Programming [33, 10, 21], generating and processing an exponential number of constraints.

Trying to solve a trade-off between summary quality, ability to process a multilingual content, and time complexity, we propose a novel summarization model that solves the approximated maximum coverage problem while attempting to cover the most important *topics* of a document set that are being ranked using a Tensor Decomposition technique. We use *tensor* as a joint representation model for three components of a document set—terms, topics, and documents—merging these components together by associations between them.

Tensor representation gives an $N$-way *connection* between its $N$ dimensions. Tensor factorization analyzes $N$ factors *simultaneously*, without the need to decide the rank of which data type precedes, like other approaches demand. As a result, we get ranking for topics, terms and documents after *single* computation performed in polynomial time. Existing internal dependencies between all three dimensions are naturally considered.

Tensor decomposition has already been used in many applicable areas [19], including summarization. For example, in [13], the authors attempted to solve the problem of comments-oriented document summarization. They focused on extracting document sentences scored to bias keywords derived from comments, while the comments, themselves, were ranked by a tensor decomposition. In this work, the authors decided that the rank of comments took precedence.

In [23] the authors summarized a document set by an extended TTI (Tensor Term Importance) model, where a term-sentence-document tensor was decomposed to highlight the important terms in each document. Then, important sentences were identified as a function of contained terms. In that study, the authors decided that rank of terms took precedence over rank of sentences. Moreover, because similar sentences often occurred in related documents of a summarized set, the authors avoided extracting redundant sentences after their ranking.

In our approach, we consider similar content before representation building and reduce the dimensionality of the tensor by using topics (clusters of sentences) instead of sentences as one of the dimensions.

Various works used sentence clustering for generating topics at some stage of a summarization pipeline [6], but none used topics as a dimension in a tensor representation of a document set. Another innovation of our approach is simultaneous ranking of multiple data types, undertaken without deciding which data type takes precedence.

In general, rank-preserving tensor decomposition is NP-hard. HOSVD (Tucker) decomposition is an approximation of rank-preserving decomposition, and as such has polynomial running time. Basic implementation of HOSVD includes computing several SVD matrix decompositions, each with an overall cost of $O(mn^2)$, where $m$ and $n$ are matrix dimensions (see [29]).

Various algorithms that improve the initially costly HOSVD algorithm exist. See [1] for a comprehensive review and efficient algorithms for $3^{rd}$-order tensors.

We measure information coverage by diversity of related topics and use the tensor decomposition technique for ranking the sentence clusters (or topics). A summary from sentences of top ranked topics is compiled in a greedy manner.

## 3 Our Method

### 3.1 Problem Setting

Extractive summarization aims at ranking text units and extracting those ranked highest into a summary. Multi-document summarization deals with a set of topic-related documents, where one summary per each set is created. In this work we introduce a method for extractive multi-document summarization, where the most informative sentences are extracted into a summary. Formally speaking, given a set of $n$ topic-related documents $D_i, 0 \leq i \leq n$, where each document $D_i$ is composed of sentences $S_{i,j}, 1 \leq j \leq k_i$, and each sentence contains $1 \leq T_{i,j,k} \leq p_{i,j}$ meaningful words (or terms), we need to find a subset $S_{i_1,j_1}, ..., S_{i_k,j_k}$ of sentences such that

1. these sentences cover the most important topics in the given document set;
2. there are at most $N$ terms in the chosen sentences;
3. redundant information within the subset of selected sentences is minimized.

### 3.2 Overview

Our approach consists of the following steps:

1. **Preprocessing**. We perform the standard text preprocessing including sentence splitting, tokenization, stop-words removal, and stemming. Stop-words removal and stemming reduce tensor dimensionality, and are optional.
2. **Topics generation**. We cluster sentences so that semantically similar sentences that address a single *topic* are grouped together, and each cluster represents some *topic*.
3. **Representation building**. Given topics, we build a three-dimensional tensor representing terms, topics, and documents, as a single unit.
4. **Topics ranking**. Given a tensor, we use its decomposition results as rankings for terms, topics, and documents.
5. **Summary compiling**. Given a ranking of the topics, we compile the summary from sentences covering the most highly ranked topics.

Below, we describe stages two through five in detail. Because many previously published works already used clustering sentences and selecting representative sentences from the clusters for compiling a summary, the focus of our paper is on *ranking topics* via Tensor Decomposition.

### 3.3 Topics Generation

In order to (1) reduce the dimensionality of the tensor representation and (2) avoid redundant information in the extracted sentences, we perform clustering on all sentences of a document set. Because all semantically similar sentences compile one cluster, we consider it to be a topic. As it usually happens, one document set consists of several related topics.

Sentences are represented by real vectors of *tf-isf* weights (because each sentence is considered a document, tf-idf weight is naturally transformed to *term frequency - inverse sentence frequency*). Formally, for each sentence we construct a real vector $\vec{v}$, where $\vec{v}[i]$ stands for the tf-isf of its $i^{th}$ term $T_i$.

We use the Suffix Tree Clustering (STC) [34] and the Lingo [24] algorithms for clustering sentences. Different algorithms were applied on different languages, as explained in Section 4.3. Both algorithms get a set of sentences as an input and produce overlapping clusters.

The STC algorithm is an incremental and linear-time (in the size of document set) clustering and labeling algorithm that uses string matching on the suffix tree structure for finding shared common phrases of documents. STC produces overlapping non-exhaustive clusters in linear time and with high precision. The STC is highly efficient.

The Lingo algorithm first extracts frequent phrases from the input documents as topic descriptions. Next, by performing reduction of the original term-document matrix using SVD, this algorithm discovers any existing latent structure of diverse topics in the documents, and finally, it matches group descriptions with the extracted topics and assigns relevant documents to them.

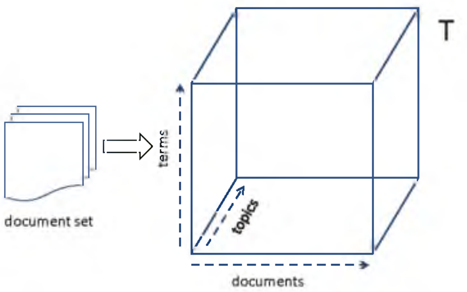

**Fig. 1.** $3$-rd order tensor representing document set.

### 3.4 Text Representation Model

An $n^{th}$*-order tensor* is defined as a multi-dimensional array with $n$ dimensions. Given a set of topic-related documents, we can build a $3^{rd}$-order tensor $T$. The first dimension of the tensor represents terms $T_1, ..., T_m$, the second dimension represents clusters of sentences $C_1, ..., C_k$, and the third dimension represents documents $D_1, ..., D_n$ in the set. The constructed tensor therefore captures all the information contained in a set of documents at the term level – each element of our tensor $t_{i,j,k}$ represents that specific information carried by the $i^{th}$ term in the $j^{th}$ cluster and the $k^{th}$ document. Using tf-idf weight as an information measure, tensor entries $t_{i,j,k}$ can be defined by the following formula:

$$\mathbf{t_{i,j,k}} = \begin{cases} \text{tf-idf}_{i,k}, & T_i \in C_j \text{ and } T_i \in D_k \\ 0, & \text{otherwise.} \end{cases}$$

Figure 1 illustrates the notion of document-topic-term tensor.

### 3.5 Topic Ranking

Based on the resulted tensor, we can measure the importance of *sentence clusters* or *topics* through tensor decomposition. Let $T$ be an $n^{th}$-order tensor over a field, whose elements are denoted by $t_{i_1 i_2 \ldots i_n}$. If $n = 1$, $T$ is a vector and if $n = 2$ $T$, is an ordinary matrix. $T$ is said to have *rank one* if it can be represented as an (outer) product of $n$ vectors $v^1, \ldots, v^n$, that is, each tensor element $t_{i_1 i_2 \ldots i_n}$ is a product of the coordinates of its respective vectors, as follows:

$$t_{i_1 i_2 \ldots i_n} = v^1_{i_1} v^2_{i_2} \ldots v^n_{i_n} \quad (1)$$

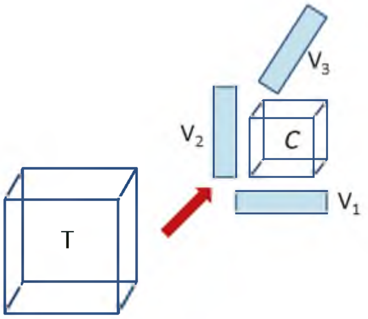

**Fig. 2.** Tensor Decomposition.

Notation is $T = v^1 \circ v^2 \circ \ldots \circ v^n$. The $rank(T)$ of a tensor is the smallest number of rank one tensors that generate $T$ as their sum, that is, a collection of $rank(T)$ $n$-tuples of vectors $v^{i,j}$ such that

$$T = \sum_{i=1}^{rank(T)} v^{i,1} \circ v^{i,2} \circ \ldots \circ v^{i,n} \quad (2)$$

Such a decomposition of a tensor into rank one tensors is hard to compute because the problem of finding the rank of a tensor is NP-hard [9]. Therefore, finding an exact tensor decomposition is a hard problem and an approximation algorithm must be used.

We use the HOSVD (High Order Singular Value Decomposition) technique that leads to orthogonal singular vectors in each dimension, assuming that latent factors are independent of each other. This decomposition is also called Tucker decomposition (introduced in [30] and later extended in [8] and [31]). Here, a tensor is approximated by a product of a smaller *core tensor* and rank one tensors. Namely,

$$T \approx \mathcal{C} \times A_1 \times A_2 \ldots \times A_n, \quad (3)$$

where $\mathcal{C}$ is the core tensor of a smaller order and $A_1, \ldots, A_n$ are the factor matrices (in most cases orthogonal) that describe rank one tensors.

In our case, tensor $T$ that lies in $\mathbb{R}^{I \times J \times K}$ has order $3$; numbers $I$, $J$, and $K$ denote its dimensions. We can therefore decompose $T$ into a product

$$T \approx \mathcal{C} \times A \times B \times C = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} c_{pqr} a_p \circ b_q \circ c_r \quad (4)$$

where $\mathcal{C}$ is the core tensor and $A, B, C$ are factor matrices. Factor matrices $A \in \mathbb{R}^P$, $B \in \mathbb{R}^Q$, and $C \in \mathbb{R}^R$ can be thought of as the principal components in each mode. The entries $c_{prq}$ of the core tensor $\mathcal{C}$ show the level of interaction between the different components.

Elementwise, the Tucker decomposition in (4) is

$$t_{ijk} \approx \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} c_{pqr}a_{ip}b_{jq}c_{kr} \qquad (5)$$

Here $P$, $Q$ and $R$ are the number of components (i.e., columns) in the factor matrices $A$, $B$, and $C$, respectively. If $P, Q, R$ are smaller than $I, J, K$, the core tensor $\mathcal{C}$ can be thought of as a compressed version of the tensor $T$. In some cases, the storage for the decomposed version of the tensor can be significantly smaller than for the original tensor.

Because we are interested in documents, sentence clusters, and terms that in combination represent the most of the data contained in the original document set, such a selection is suspected to contribute the most to tensor ranking. The decomposition in question should imply importance ranking of documents, sentence clusters, and terms. We strive to obtain both ranking and compression of initial document(s) simultaneously using the means that Tucker decomposition provides. Therefore, we seek a core tensor $\mathcal{C}$ and vectors $V_1, V_2, V_3$ so that

$$T \approx \mathcal{C} \times V_1 \times V_2 \times V_3 \qquad (6)$$

where vectors $V_i$ reflect the ranking of the entities from the corresponding dimension. In our setting, $V_1$ contains the salience scores for documents, $V_2$ contains the importance scores for terms, and $V_3$ contains the same information regarding topics in a set. Figure 2 shows the result of such decomposition.

### 3.6 Summary Compiling

Given the ranking of topics in topic vector $V_3$, we can compose a summary for a document set by extracting *centroid* sentences from clusters, starting from the higher ranked to those with a lower rank, until we reach the maximal summary length. We define the centroid sentence as the sentence with the minimal average distance to all cluster sentences. In our setting, we use cosine similarity as a distance between sentences.

If after going over all clusters the summary does not reach the maximal length, we repeat the same process extracting a next sentence with a minimal average distance from each cluster in a greedy manner, until we get a summary of a necessary length.

The selected sentences are organized first in the chronological order of their documents, and second, in order of their appearance in a document, according to [3].

## 4 Experiments

### 4.1 Experimental Setup

Our experiments aim to test the behavior of our approach on multiple languages, and to compare the performance of our approach to the baseline and to other multi-document multilingual summarizers in a multilingual domain.

The detailed descriptions of other summarizers used in our comparisons can be found in the proceedings of Text Understanding Conference (TAC) 2011.

These systems participated in the TAC MultiLing 2011 pilot [7] and were shown to be comparable to the global topline and the global baseline systems. The *global topline system–$ID10$*–uses human model summaries (thus cheating). The *global baseline system–$ID9$*–uses a bag-of-words approach to represent the documents of a topic in vector space.

The system uses the text that is most similar to the centroid (based on the cosine similarity) of the document set in the summary.

We used the method of ranking topics by their average tf-idf coverage as another baseline approach. Formally, the rank of $C_j$ standing for $j^{th}$ topic, was calculated as an average of tensor entries $t_{i,j,k} \in C_j$, as follows:

$$\frac{\sum_i^n \sum_k^m t_{i,j,k}}{|t_{i,j,k} \in C_j|}$$

DOI: 10.13053/CyS-18-3-2026

**Table 1.** Evaluation results: English corpus

| system | rouge-1 | system | rouge-2 | system | rouge-SU4 |
|---|---|---|---|---|---|
| ID10 | 0.525 | ID10 | 0.252 | ID10 | 0.273 |
| ID2 | 0.465 | ID3 | 0.174 | ID2 | 0.202 |
| **TeDeS** | 0.445 | ID2 | 0.171 | ID3 | 0.200 |
| ID4 | 0.444 | **TeDeS** | 0.155 | **TeDeS** | 0.192 |
| ID3 | 0.432 | ID4 | 0.152 | ID4 | 0.191 |
| Base | 0.428 | Base | 0.135 | Base | 0.176 |
| ID5 | 0.411 | ID5 | 0.136 | ID5 | 0.175 |
| ID1 | 0.406 | ID1 | 0.122 | ID1 | 0.160 |
| ID7 | 0.396 | ID8 | 0.121 | ID8 | 0.157 |
| ID8 | 0.387 | ID9 | 0.110 | ID9 | 0.148 |
| ID9 | 0.381 | ID6 | 0.107 | ID6 | 0.146 |
| ID6 | 0.355 | ID7 | 0.097 | ID7 | 0.145 |

### 4.2 Experimental Data

We performed experiments on a MultiLing [7] dataset using English, Hebrew, and Arabic languages. The MultiLing corpus consists of 10 document sets, with 10 documents each set, in seven languages. Original news articles in English were taken from WikiNews[1], organized into 10 sets, manually translated to Arabic, Czech, French, Greek, Hebrew, and Hindi, and then summarized.

### 4.3 External Tools

For sentence clustering we adapted the STC and the Lingo3G algorithms from the Carrot2 tool [32]. The STC algorithm was applied on English documents, and the Lingo3G algorithm was applied on the Hebrew and Arabic languages. We used different clustering methods for different languages because the best performing algorithm (STC) works for English texts only.

For stemming, the Porter stemmer [16, 25], the morphological analyzer [15, 14] and the Arabic stemmer [17] were used in the English, Hebrew, and Arabic corpora, respectively.

The Matlab Tensor Toolbox [2] toolkit was utilized for performing mathematical operations on tensors, including decomposition.

### 4.4 Evaluation Metrics

For the quality assessment, we measured Rouge-1,2, and SU4 (recall-based) scores [20] and compared the results of our method with other systems participating in the MultiLing pilot.

### 4.5 Experimental Results

The results of our experiments are shown in Tables 1,3, and 2. Our method is denoted as TeDeS (Tensor Decomposition-based Summarizer). Our baseline method, which did not participate in the MultiLing pilot, is denoted by Base. In Tables 1,3, and 2, below, we highlighted by grey all systems with performance that is statistically indistinguishable from that of TeDeS, according to the paired t-test with 95% confidence interval. As can be seen from Table 1, in the English dataset our method outperforms 8 out of 10 other methods in terms of Rouge-1, and 7 systems in terms of Rouge-2 and Rouge-SU4. Also, our method significantly outperforms our baseline method in all metrics. Two–$ID2$ and $ID3$–systems, except a topline, outperformed our summarizer in terms of Rouge-2 and Rouge-SU4, and only $ID2$ in terms of Rouge-1. Performances of TeDeS and $ID4$ are statistically indistinguishable.

In Arabic, our method outperforms 7, 5, and 6 out of 9 systems in terms of Rouge-1, Rouge-2, and Rouge-SU4, respectively. The difference in performance of TeDeS and our baseline is considered

[1] http://en.wikinews.org/wiki/

**Table 2.** Evaluation results: Arabic corpus

| system | rouge-1 | system | rouge-2 | system | rouge-SU4 |
|---|---|---|---|---|---|
| ID10 | 0.788 | ID10 | 0.570 | ID10 | 0.629 |
| Base | 0.748 | ID4 | 0.501 | ID4 | 0.567 |
| ID4 | 0.745 | ID2 | 0.491 | Base | 0.558 |
| **TeDeS** | 0.744 | ID3 | 0.483 | ID2 | 0.552 |
| ID2 | 0.737 | Base | 0.483 | **TeDeS** | 0.550 |
| ID1 | 0.725 | **TeDeS** | 0.478 | ID3 | 0.540 |
| ID8 | 0.725 | ID8 | 0.460 | ID8 | 0.536 |
| ID7 | 0.721 | ID1 | 0.447 | ID1 | 0.526 |
| ID9 | 0.704 | ID7 | 0.442 | ID7 | 0.523 |
| ID3 | 0.700 | ID9 | 0.440 | ID9 | 0.514 |
| ID6 | 0.657 | ID6 | 0.433 | ID6 | 0.493 |

not significant. Also, $ID4$, $ID3$, and $ID2$ perform in a manner that is similar to TeDeS in terms of one of Rouge metrics. The results are shown in Table 2.

In Hebrew (Table 3) our method does not perform well, outperforming only $4$, $3$, and $2$ out of $7$[2] systems in terms of Rouge-$1$, Rouge-$2$, and Rouge-SU$4$, respectively. Generally, experimental results derived from the Hebrew domain are quite different from those obtained on the other two languages, and were observed to be inconsistent. We attribute this phenomenon principally to linguistic and translation differences, rather than to a defect in the TeDeS method.[3]

Generally, three systems–$ID2$, $ID3$, and $ID4$–outperformed TeDeS in many cases. Below, we briefly describe those systems.

CLASSY–$ID2$–summarizer [4] uses a Naive Bayes model for term scoring. It scores sentences by the normalized number of "summary content terms". A non-redundant subset of high scoring sentences is chosen, using non-negative matrix factorization. In order to achieve the desired summary length, it uses a branch and bound algorithm to approximately solve a knapsack problem.

JRC–$ID3$–is an LSA-based summarizer [27] that includes temporal analysis for improving sentence ordering, detection of update information, and dealing with the WHEN aspect. This summarizer also compresses and reconstructs sentences.

LIF–$ID4$–system [11] modifies a system based on the Maximal Marginal Relevance (MMR) algorithm in order to remove or minimize dependencies on language during sentence segmentation, word segmentation, and stop-words removal. In particular, sentence segmentation is replaced by a crude heuristic, words are replaced by character n-grams, and as such, there is no need to rely on a formal stop-words list.

## 5 Conclusions and Perspectives

In this paper we introduced a novel method for multi-document, multilingual summarization based on a Tensor Decomposition technique. Our method represents a set of related documents as a $3^{rd}$-order tensor with dimensions standing for terms, documents, and topics. The rank of topics is then retrieved by a Tensor Decomposition, and a summary is compiled in attempt to cover the most important topics. This is accomplished by extracting the most representative sentences from the most highly ranked topics, using a greedy approach. The proposed method is unsupervised and can be easily applied to multiple languages.

The experiments show that our method has a good performance on English and Arabic. In Hebrew, our method outperforms $3$ out of $7$ systems.

[2] Not all systems performed their evaluations on all seven languages. Only $7$ systems were evaluated on the Hebrew corpus, and $9$ systems were evaluated on the Arabic corpus.

[3] In particular, the Hebrew corpus was produced by volunteers that are not professional translators or linguists.

**Table 3.** Evaluation results: Hebrew corpus

| system | rouge-1 | system | rouge-2 | system | rouge-SU4 |
|---|---|---|---|---|---|
| ID10 | 0.565 | ID10 | 0.173 | ID10 | 0.292 |
| ID4 | 0.502 | ID3 | 0.123 | ID4 | 0.199 |
| ID2 | 0.371 | ID7 | 0.110 | ID2 | 0.162 |
| **TeDeS** | 0.356 | ID2 | 0.093 | ID3 | 0.158 |
| Base | 0.345 | Base | 0.078 | ID9 | 0.156 |
| ID3 | 0.338 | **TeDeS** | 0.072 | **TeDeS** | 0.131 |
| ID9 | 0.331 | ID9 | 0.070 | ID7 | 0.124 |
| ID7 | 0.296 | ID4 | 0.063 | Base | 0.094 |
| ID1 | 0.250 | ID1 | 0.053 | ID1 | 0.090 |

In addition, because HOSVD (Tucker) decomposition has polynomial running time, the introduced method is efficient.

In the future, we plan to evaluate our method by testing it with more languages. We expect to extend our model to other IR domains, where simultaneous ranking of multiple data types is useful. We anticipate being able to optimize the current model by further dimensionality reduction, where the terms dimension can be replaced by "topic" words, using Topic Modeling.

## Acknowledgments

Authors are grateful to Igor Vinokur for the plugin implementation and technical support.

## References

1. **Badeau, R. & Boyer, R.** (**2008**). Fast multilinear singular value decomposition for structured tensors. *SIAM. J. Matrix Anal. and Appl.*, 30(3), 1008–1021.
2. **Bader, B. W., Kolda, T. G., et al.** (**2012**). Matlab tensor toolbox version 2.5.
3. **Barzilay, R., Elhadad, N., & McKeown, K. R.** (**2001**). Sentence ordering in multidocument summarization. In *Proceedings of the First International Conference on Human Language Technology Research*. 1–7.
4. **Conroy, J. M., Schlesinger, J. D., Kubina, J., Rankel, P. A., & O'Leary, D. P.** (**2011**). CLASSY 2011 at TAC: Guided and Multi-lingual Summaries and Evaluation Metrics. In *Proceedings of TAC 2011*.
5. **Evans, D. K., Mckeown, K., & Klavans, J. L.** (**2005**). Similarity-based multilingual multi-document summarization. *IEEE Transactions on Information Theory*, 49.
6. **Filatova, E. & Hatzivassiloglou, V.** (**2004**). Event-based extractive summarization. In *In Proceedings of ACL Workshop on Summarization*. 104–111.
7. **Giannakopoulos, G., El-Haj, M., Favre, B., Litvak, M., Steinberger, J., & Varma, V.** (**2011**). TAC 2011 MultiLing Pilot Overview. In *TAC 2011: Proceedings of Text Analysis Conference*.
8. **Gulliksen, H. & Frederiksen, N.** (**1964**). The extension of factor analysis to three-dimensional matrices. *Contributions to Mathematical Psychology*.
9. **Hastad, J.** (**1990**). Tensor rank is np-complete. *Journal of Algorithms*, 11, 644–654.
10. **Hitoshi Nishikawa, Y. M., Takaaki Hasegawa & Kikui, G.** (**2010**). Opinion Summarization with Integer Linear Programming Formulation for Sentence Extraction and Ordering. In *Coling 2010: Poster Volume*. 910–918.
11. **Hmida, F. & Favre, B.** (**2011**). LIF at TAC MultiLing: Towards a Truly Language Independent Summarizer. In *Proceedings of TAC 2011*.
12. **Honarpisheh, M. A., Ghassem-Sani, G., & Mirroshandel, G.** (**2008**). A multi-document multilingual automatic summarization system. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP 2008)*. 733–738.
13. **Hu, M., Sun, A., & peng Lim, E.** (**2008**). Comments-oriented document summarization: Understanding documents with readers feedback. In *In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR 2008. ACM.*

14. **Itai, A., Jacob, Y., & Chen, G. (2003).** MILA: Knowledge Center for Processing Hebrew.
15. **Itai, A. & Wintner, S. (2008).** Language resources for Hebrew. *Language Resources and Evaluation*, 42(1), 75–98.
16. **Jones, K. S. & Willet, P. (1997).** *Readings in Information Retrieval*. San Francisco: Morgan Kaufmann. ISBN 1-55860-454-4.
17. **Khoja, S. (2001).** Arabic stemmer.
18. **Khuller, S., Moss, A., & Naor, J. S. (1999).** The budgeted maximum coverage problem. *Information Precessing Letters*, 70(1), 39–45.
19. **Kolda, T. G. & Bader, B. W. (2007).** Tensor decompositions and applications. Technical report, Sandia National Laboratories.
20. **Lin, C.-Y. (2004).** Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*. 25–26.
21. **Makino, T., Takamura, H., & Okumura, M. (2011).** Balanced coverage of aspects for text summarization. In *TAC 2011: Proceedings of Text Analysis Conference*.
22. **Mani, I. & Maybury, M. (1999).** *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA.
23. **Manna, S., Petres, Z., & Gedeon, T. (2009).** Tensor term indexing: An application of HOSVD for document summarization. In *4th International Symposium on Computational Intelligence and Intelligent Informatics, 2009. ISCIII '09*. 135–141.
24. **Osinski, S., Stefanowski, J., & Weiss, D. (2004).** Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition. In *Intelligent Information Systems*. 359–368.
25. **Porter, M. (2006).** The porter stemming algorithm.
26. **Saggion, H. (2006).** Multilingual multidocument summarization tools and evaluation. In *Proceedings of the International Conference on Language Resources and Evaluation*. 1312–1317.
27. **Steinberger, J., Kabadjov, M., Steinberger, R., Tanev, H., Turchi, M., & Zavarella, V. (2011).** JRC Participation at TAC 2011: Guided and Multilingual Summarization Tasks. In *Proceedings of TAC 2011*.
28. **Takamura, H. & Okumura, M. (2009).** Text summarization model based on maximum coverage problem and its variant. In *EACL 2009: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. 781–789.
29. **Trefethen, L. N. & Bau, D. (1997).** *Numerical linear algebra*. Philadelphia: Society for Industrial and Applied Mathematics.
30. **Tucker, L. (1963).** Implications of factor analysis of three-way matrices for measurement of change. *Problems in Measuring Change*, 122–137.
31. **Tucker, L. (1966).** Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31, 279–311.
32. **Weiss, D. & Osinski, S. (2004).** Carrot$^2$ open source search results clustering engine. `http://search.carrot2.org`.
33. **Woodsend, K. & Lapata, M. (2010).** Automatic Generation of Story Highlights. In *ACL 2010: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. 565–574.
34. **Zamir, O. & Etzioni, O. (1998).** Web Document Clustering: A Feasibility Demonstration. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 46–54.

**Marina Litvak** obtained a Ph.D. in Information Systems Engineering from Ben-Gurion University of the Negev in 2010. She is currently a faculty member at Department of Software Engineering of Shamoon College of Engineering in Beer Sheva, Israel. Her research interests include information retrieval, text mining, automated summarization, social networks analysis, and recommender systems.

**Natalia Vanetik** obtained a Ph.D. in Computer Science from Ben-Gurion University of the Negev in 2009. She is currently a faculty member at Department of Software Engineering of Shamoon Academic College of Engineering in Beer Sheva, Israel. Her research interests include data mining, combinatorial optimization, text mining and text analysis and biological data mining.

*Article received on 31/12/2013; accepted on 12/02/2014.*