Tensor Term Indexing: An application of HOSVD for Document Summarization

Sukanya Manna[1], Zoltán Petres[1,2], and Tom Gedeon[1]

[1]School of Computer Science, The Australian National University, ACT 0200, Australia
[2]Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary
E-mail: {sukanya.manna,tom.gedeon}@anu.edu.au, petres@sztaki.hu

Abstract— **In this paper, a new method for text summarization is proposed by using an extended version of the Tensor Term Importance (TTI) model. This method summarizes documents by extracting important sentences from a document. It improves the per document summarization efficiency by incorporating additional information of the whole document set referring to the same topic (or coherent documents).**

The basic idea of this approach is to represent the whole document set in a uniform form, in the term-sentence-document tensor, and to use higher-order singular value decomposition (HOSVD) to highlight the important terms in each document. Here, we present two different methods of summarization. In the first method, the sentences having the highly weighted terms are extracted as the important sentences representing the document. The important sentences identified by selecting those that contains more from the important terms. The second model uses a so-called super sentence and uses that to extract other sentences having high similarity with it.

Unlike in Latent Semantic Analysis (LSA) where SVD is applied for compressing the sparse term-document matrix and defining latent semantic links between terms, in TTI SVD is used to reduce noise and to highlight the important term-document relations in the document.

Our evaluation results show that our TTI based methods are more similar to human generated summaries than other automated summarizers which work on single documents at a time.

I. Introduction

Automatic text summarization is the technique which automatically creates an abstract or summary of a text. The technique has been developed for many years [1], [2], and [3]. According to Hovy and Lin [4] there are two ways to view text summarization either as text extraction or text abstraction. Text extraction means to extract pieces of an original text on a statistical basis or with heuristic methods and put them together into a shorter text with the same information content. Sometimes, the extracted fragments are post-edited, for example by deleting subordinate clauses or joining incomplete clauses to form complete clauses [5], [6]. Text abstraction is to parse the original text in a linguistic way, interpret the text and find new concepts to describe the text and then generate a new shorter text with the same information content. This is in many aspects similar to what human abstractors do when writing an abstract, using surface level information like headings, key phrases, positions and so on [7], [8], [9].

There are some language independent approaches which also perform summarization. Among them Gong and Liu [10] and the approach used by Yeh et. al, [11] both use Latent Semantic Analysis for sentence extraction. TextRank is another approach which employs iterative graph-based ranking algorithms to encode the cohesive structures of a text [12], [13]. The approach presented by Gong and Liu creates text summaries by ranking and extracting sentences from the original documents, where they use LSA to create *synonym sets* or rather *semantic sets*, which are used to pinpoint the topically central sentences. In the approach by Yeh et al., they used LSA to derive a sematic matrix of a document. Based on this, semantic sentence representations are used to construct a text relationship map [14] for interpreting conceptual structures of a document.

The Knowledge Management (KM) systems from SRA International, Inc.[1] extracts summarization features using morphological analysis, name tagging and co-references resolution. They use a machine learning technique to determine the optimal combination of these features in combination with statistical information from the corpus to identify the best sentences to include in a summary.

In this paper, we propose an extended model for document summarization based on Tensor Term Importance (TTI). This uses SVD to extract the important terms unlike Latent Semantic Analysis (LSA) [15], in which the main goal for its usage is to reduce the dimensions for finding the semantic relations. In our proposed method we use SVD for a different reason. Basically it is used to reduce noise and then transform the low noise result back to the original tensor structure. The most important sentence, according to our first model is the one which contains the important keywords and in our second model, the most important sentence is the one which has high similarity with the super sentence (i.e. a virtual central sentence, which consists of all the important words).

II. Preliminaries

A. Term-Sentence Matrix and Term-Sentence-Document Tensor

In this study, term-sentence matrices and term-sentence-document tensors are used to represent a document and set of documents, respectively.

[1]http://www.SRA.com

978-1-4244-5382-5/09/$26.00 ©2009 IEEE

a) Term-Sentence Matrix: Let D be a document, T ($|T| = N$) be the set of terms in D, and S ($|S| = M$) be the set of sentences in D. An $N \times M$ term-document matrix, A, is constructed as Eq. (1), where S_i indicated a sentence and T_i indicates a term. In our work, only nouns and verbs are taken into account in that they carry essential information about the meaning of a sentence, and the stop words are neglected.

$$A = \begin{array}{c|cccc} & S_1 & S_2 & \dots & S_N \\ \hline T_1 & a_{1,1} & a_{1,2} & \dots & a_{1,N} \\ T_2 & a_{2,1} & a_{2,2} & \dots & a_{2,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_M & a_{M,1} & a_{M,2} & \dots & a_{M,N} \end{array} \quad (1)$$

In A, $a_{i,j}$ is defined as the term frequency, i.e. the number of occurrence of term T_i in sentence S_j.

b) Term-Sentence-Document Tensor: We represent the collection of documents in a similar way it is done for a document in a term-sentence matrix. The collection is represented in a tensor where the dimensions represent the term, sentences, and documents, respectively. Thus, let C be a collection of documents, D ($|D| = O$) be a document of the collection, T_i ($|T_i| = N_i$) be the set of terms in D_i, and S_i ($|S_i| = M_i$) be the set of sentences in D_i. A term-sentence-document tensor, $\mathcal{A}$, is constructed, where the terms are considered as the superset of the terms in each document ($T = \{T_1, T_2, \dots, T_O\}$, $|T| = N$), the number of sentences is $M = \max M_i$. Accordingly, in the $T \times S \times O$-size tensor $\mathcal{A}$, $a_{i,j,k}$ is defined as the number of occurrence of term T_i in sentence S_j of document D_k.

Before the construction of Term-Sentence Matrix or Term-Sentence-Document Tensor, a preprocessing on the input document collection is performed to reduce the noise and improve the method's performance. This preprocessing includes the tokenization of the document collection, word stemming, and removal of stop and common words.

B. Singular Value Decomposition (SVD) and Higher Order Singular Value Decomposition

In this subsection the definition and important properties of SVD and HOSVD [16] is given. The tensor notation, some definitions, and terminology is based on Lathauwer's work, detailed in [16].

c) Matrix SVD: Every real $(I_1 \times I_2)$-matrix $\mathbf{F}$ can be written as the product

$$\begin{aligned} \mathbf{F} &= \mathbf{U}_{(1)} \cdot \mathbf{S} \cdot \mathbf{V}_{(2)}^T = \mathbf{S} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{V}_{(2)} \\ &= \mathbf{S} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{U}_{(2)} = \mathbf{S} \overset{2}{\underset{n=1}{\boxtimes}} \mathbf{U}_{(n)}, \end{aligned}$$

in which

1) $\mathbf{U}_{(1)} = \left(\mathbf{u}_1^{(1)} \mathbf{u}_2^{(1)} \cdots \mathbf{u}_{I_1}^{(1)}\right)$ is a unitary $(I_1 \times I_1)$-matrix,
2) $\mathbf{U}_{(2)} = \left(\mathbf{u}_1^{(2)} \mathbf{u}_2^{(2)} \cdots \mathbf{u}_{I_1}^{(2)}\right)$ is a unitary $(I_2 \times I_2)$-matrix,
3) $\mathbf{S}$ is an $(I_1 \times I_2)$-matrix with the properties of
   a) pseudodiagonality:

$$\mathbf{S} = \operatorname{diag}\left(\sigma_1, \sigma_2, \dots, \sigma_{\min(I_1, I_2)}\right), \quad (2)$$

   b) ordering:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(I_1, I_2)} \geq 0. \quad (3)$$

The σ_i are singular values of $\mathbf{F}$ and the vectors $\mathbf{u}_i^{(1)}$ and $\mathbf{u}_i^{(2)}$ are, respectively, an ith left and an ith right singular vector.

The number of non-zero singular values σ_i equals to the rank of matrix $\mathbf{F}$.

d) Tensor unfolding: The starting point in the derivation of a multilinear singular value decomposition for tensors, as multi-dimensional matrices, is to consider an appropriate generalization of the link between the column (row) vectors and the left (right) singular vectors of a matrix. In order to formalize this idea, we define the matrix representations of the tensor in which all the column (row, ...) vectors are stacked one after the other in the following way:

Definition 1 (n-mode matrix of tensor $\mathcal{A}$): Assume an Nth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$. The matrix $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-1})}$ contains the element $a_{i_1, i_2, \dots, i_N}$ at the position with row number i_n and column number equal to:

$$\begin{aligned} &(i_{n+1} - 1) I_{n+2} I_{n+3} \cdots I_N I_1 I_2 \cdots I_{n-1} + \\ &+ \quad (i_{n+2} - 1) I_{n+3} I_{n+4} \cdots I_N I_1 I_2 \cdots I_{n-1} + \cdots + \\ &+ \quad (i_N - 1) I_1 I_2 \cdots I_{n-1} + (i_1 - 1) I_2 I_3 \cdots I_{n-1} + \\ &+ \quad (i_2 - 1) I_3 I_4 \cdots I_{n-1} + \cdots + i_{n-1}. \end{aligned}$$

Remark 1: The ordering of the column vectors can be arbitrarily determined. The only important thing is that in all cases the same ordering and reordering must be used systematically later on. In general, the rth column of n-mode matrix $\mathbf{A}_{(n)}$ is equivalent to the $I_1, I_2, \dots, I_{n-1}, I_{n+1}, \dots, I_N$-th vector of dimension n, where

$$r = \operatorname{ordering}(i_1, i_2, \dots, i_{n-1}, i_{n+1}, \dots, i_N).$$

Figure 1 shows an example for the n-mode matrix of a 3rd-order tensor.

e) Higher Order SVD, HOSVD: Every real $(I_1 \times I_2 \times \cdots \times I_N)$-tensor $\mathcal{A}$ can be written as the product

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{U}_{(2)} \times_3 \cdots \times_N \mathbf{U}_{(N)} = \mathcal{S} \overset{N}{\underset{n=1}{\boxtimes}} \mathbf{U}_{(n)}, \quad (4)$$

in which

1) $\mathbf{U}_{(n)} = \left(\mathbf{u}_1^{(n)} \mathbf{u}_2^{(n)} \cdots \mathbf{u}_{I_n}^{(n)}\right)$, $n = 1 \dots N$ is a unitary $(I_n \times I_n)$-matrix,
2) $\mathcal{S}$ is a real $(I_1 \times I_2 \times \cdots \times I_N)$-tensor of which the subtensors $\mathcal{S}_{i_n = \alpha}$ obtained by fixing the nth index to α, have the properties of
   a) all-orthogonality: two subtensors $\mathcal{S}_{i_n = \alpha}$ and $\mathcal{S}_{i_n = \beta}$ are orthogonal for all possible values of n, α and β subject to $\alpha \neq \beta$:

$$\langle \mathcal{S}_{i_n = \alpha}, \mathcal{S}_{i_n = \beta} \rangle = 0, \text{ when } \alpha \neq \beta, \quad (5)$$

   b) ordering:

$$\|\mathcal{S}_{i_n = 1}\| \geq \|\mathcal{S}_{i_n = 2}\| \geq \cdots \geq \|\mathcal{S}_{i_n = I_n}\| \geq 0, \quad (6)$$

   for all possible values of n.

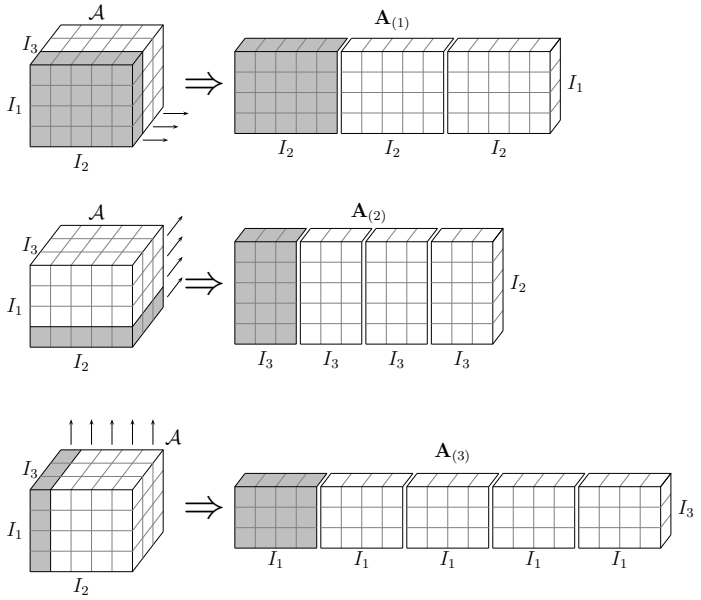

Fig. 1. Illustration of 3-mode matrices of a 3rd-order tensor $\mathcal{A}$

The Frobenius-norms $\|\mathcal{S}_{i_n=i}\|$, symbolized by $\sigma_i^{(n)}$, are n-mode singular values of $\mathcal{A}$ and the vector $\mathbf{u}_i^{(n)}$ is an ith n-mode singular vector. The decomposition is visualized for third-order tensors in Figure 2.

Note that the HOSVD uniquely determines tensor $\mathcal{S}$, but the determination of matrices $\mathbf{U}_{(n)}$ may not be unique if there are equivalent singular values at least in one dimension.

f) Approximation trade-off by HOSVD: If we discard non-zero singular values (not only the zero ones) and the corresponding singular vectors, then the decomposition only results an approximation of tensor $\mathcal{S}$ with the following property.

Assume the HOSVD of tensor $\mathcal{A}$ is given, and the n-mode rank of $\mathcal{A}$ is R_n $(1 \leq n \leq N)$. Let us define $\hat{\mathcal{A}}$ by changing the corresponding elements of singular values $\sigma_{I'_n+1}^{(n)}, \sigma_{I'_n+2}^{(n)}, \ldots \sigma_{R_n}^{(n)}$ of tensor $\mathcal{S}$ to zero, for a given $I'_n < R_n$. In this case

$$\gamma = \|\mathcal{A} - \hat{\mathcal{A}}\|^2 \leq \sum_{i_1=I'_{1+1}}^{R_1} \left(\sigma_{i_1}^{(1)}\right)^2 + \sum_{i_2=I'_{2+1}}^{R_2} \left(\sigma_{i_2}^{(2)}\right)^2 + \\ + \cdots + \sum_{i_N=I'_{N+1}}^{R_N} \left(\sigma_{i_N}^{(N)}\right)^2.$$

This property is the Nth-order generalization of the connection between the singular value decomposition of a matrix and its best, lower ranked matrix approximation (in the sense of least square).

III. Tensor Term Indexing

Tensor Term Indexing (TTI) is a novel method to extract the important terms of a document in a relatively small coherent document collection. Term extraction approaches tend to decrease their discriminative power when many documents refer to the same topic and fail to identify the topic words. For subjective analysis of documents, these topic words are essential for applications like intelligent document analysis which are more context oriented. Our main motivation is to extract the significant terms from a single document as well as a collection of coherent documents referring to a particular topic without losing discrimination power, and select the meaningful sentences for summarization and abstraction based on the significant terms.

Tensor Term Indexing represents the document in Term-Sentence-Document Tensor format as defined in the Section II. This description enables the method to analyze intra- and inter-document term relations at the same time. The basic idea behind TTI is to form a new tensor that has the same structure as the original, but is an approximation of the original which describes the documents in a condensed way. We then weight the terms for single documents by summing the nonzero SVD generated values of the new tensor across the sentences of each documents. Similarly, for finding the weights on the whole document collection, we "unfold" the new tensor, and perform the same process to find the importance of words across all the documents.

A. Derivation of TTI

TTI uses the lower rank approximation technique to reduce the noise by eliminating anecdotal terms, to mitigate synonymy by expecting to merge the dimensions associated with terms that have similar meanings, and to mitigates polysemy, since components of polysemous words that point in the "right" direction are added to the components of words that share a similar meaning. Conversely, components that point in other directions tend to either simply cancel out, or, at worst, to be

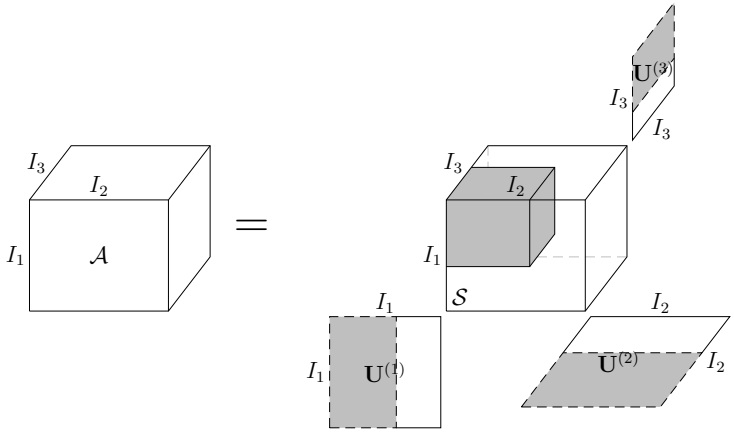

Fig. 2. Visualization of the HOSVD for a third-order tensor

smaller than components in the directions corresponding to the intended sense.

Let $\mathbf{X}$ be a term-sentence matrix as defined in Section II. Now a row in this matrix will be a vector corresponding to a term, giving its relation to each sentence:

$$\mathbf{t}_i^T = \begin{bmatrix} x_{i,1} & \dots & x_{i,n} \end{bmatrix}$$

Likewise, a column in this matrix will be a vector corresponding to a sentence, giving its relation to each term:

$$\mathbf{s}_j = \begin{bmatrix} x_{1,j} \\ \vdots \\ x_{m,j} \end{bmatrix}$$

Now the dot product $\mathbf{t}_i^T \mathbf{t}_p$ between two term vectors gives the correlation between the terms over the sentences. The matrix product $\mathbf{X}\mathbf{X}^T$ contains all these dot products. Element (i, p) (which is equal to element (p, i)) contains the dot product $\mathbf{t}_i^T \mathbf{t}_p$ ($= \mathbf{t}_p^T \mathbf{t}_i$). Likewise, the matrix $\mathbf{X}^T\mathbf{X}$ contains the dot products between all the sentence vectors, giving their correlation over the terms: $\mathbf{s}_j^T \mathbf{s}_q = \mathbf{s}_q^T \mathbf{s}_j$.

Now decompose $\mathbf{X}$ such that $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices and $\mathbf{S}$ is a diagonal matrix. This is called matrix SVD:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

The matrix products giving us the term and sentence correlations then become

$$\begin{aligned} \mathbf{X}\mathbf{X}^T &= (\mathbf{U}\mathbf{S}\mathbf{V}^T)(\mathbf{U}\mathbf{S}\mathbf{V}^T)^T = (\mathbf{U}\mathbf{S}\mathbf{V}^T)(\mathbf{V}^{T^T}\mathbf{S}^T\mathbf{U}^T) \\ &= \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}^T\mathbf{U}^T = \mathbf{U}\mathbf{S}\mathbf{S}^T\mathbf{U}^T \\ \mathbf{X}^T\mathbf{X} &= (\mathbf{U}\mathbf{S}\mathbf{V}^T)^T(\mathbf{U}\mathbf{S}\mathbf{V}^T) = (\mathbf{V}^{T^T}\mathbf{S}^T\mathbf{U}^T)(\mathbf{U}\mathbf{S}\mathbf{V}^T) \\ &= \mathbf{V}\mathbf{S}\mathbf{U}^T\mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T \end{aligned}$$

Since $\mathbf{S}\mathbf{S}^T$ and $\mathbf{S}^T\mathbf{S}$ are diagonal we see that U must contain the eigenvectors of $\mathbf{X}\mathbf{X}^T$, while $\mathbf{V}$ must be the eigenvectors of $\mathbf{X}^T\mathbf{X}$. Both products have the same non-zero eigenvalues, given by the non-zero entries of $\mathbf{S}\mathbf{S}^T$, or equally, by the non-zero entries of $\mathbf{S}^T\mathbf{S}$.

It turns out that when you select the k largest singular values, and their corresponding singular vectors from $\mathbf{U}$ and $\mathbf{V}$, you get the rank k approximation to $\mathbf{X}$ with the smallest error (Frobenius norm) (see the approximation trade-off property of SVD, HOSVD). The amazing thing about this approximation is that not only does it have a minimal error, but it translates the term and sentence vectors into a concept space. The vector $\hat{\mathbf{t}}_i$ then has k entries, each giving the occurrence of term i in one of the k concepts. Likewise, the vector $\hat{\mathbf{s}}_j$ gives the relation between sentence j and each concept. We write this approximation as

$$\mathbf{X}_k = \mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^T$$

The similar derivation can be done for term-sentence-document tensor $\mathcal{X}$ using HOSVD to get its rank k approximation as:

$$\mathcal{X}_{\text{TTI}} = \mathcal{S}_k \bigboxtimes_{n=1}^{N} \mathbf{U}_{(n)}^k, \tag{7}$$

The term-sentence-tensor $\mathcal{X}_{\text{TTI}}$ gives an emphasized description of the original document collection. The HOSVD-based lower rank approximation has reduced the noise, merged terms with similar meanings, and adjusted the terms per document significance by considering its significance in other documents in the collection.

B. Document summarization using important terms of TTI

TTI is an efficient tool to rank the terms in a document and extract the most significant ones to give a summary. We can weight the sentences in a document by measuring the contribution of significant terms in each sentence. The more the contribution of significant terms in a document, i.e. the sentence contains more important terms, the better candidate to summarize the document. We define the following measure for a sentence s in document i:

$$w_{i,s}^{\text{IT}} = |\{\forall j, x_{i,s,j} : x_{i,s,j} > 0\}| \sum_{j=1\dots J} x_{i,s,j}, \tag{8}$$

where $x_{i,s,j}$ is an item of Tensor Term Index $\mathcal{X}_{\text{TTI}}$, J is the number of terms in the document collection, and $|\{\forall j, x_{i,s,j} :$

$x_{i,s,j} > 0\}|$ gives the number of terms, where the term has a value higher than zero.

The higher the weight $w_{i,s}^{\text{IT}}$ of a sentence in a document, the more important it is. Therefore, a descending ordering lists the sentences in the order of good candidates for summarization.

C. Document summarization using the super sentence of TTI

The ultimate method for text summarization would be to identify all the significant terms of a document, put them together and compose a proper sentence with them. This cannot be achieved properly with the existing solutions, however the following method based on TTI offers a solution that tries to mimic this approach.

The 1-mode matrix of $\mathcal{X}_{\text{TTI}}$ can be decomposed by HOSVD as

$$\mathbf{X}_{\text{TTI}}^{(1)} = \mathbf{U}^{(1)}\mathbf{S}^{(1)}{\mathbf{V}^{(1)}}^{T},$$

where matrices $\mathbf{U}$ and $\mathbf{V}$ are unitary, and $\mathbf{S}$ is a diagonal matrix containing the singular values is descending order. The matrix $\mathbf{U}$ does the linear transformation from the original document-sentence-term space into the conceptual term description space, where the content is represented in a condensed, and abstracted way. For each singular value in $\mathbf{S}$ there are corresponding row and column vectors, the singular vectors, in $\mathbf{U}$ and $\mathbf{V}^T$, respectively. The descending order and the size of the singular values show the contribution of each singular vector to the approximation. Based on this observation, we can compose a vector, a so-called super sentence, that contains all the terms of the whole document set, and each term has a weight in this super sentence that indicates its significance for summarization.

The super sentence is calculated as

$$\mathbf{ss} = \sum_{i=1\ldots k} s_{k,k}^{(1)}\mathbf{u}_k^{(1)},$$

where $s_{k,k}$ is the kth singular value in matrix $\mathbf{S}$ and $\mathbf{u}_k$ is the kth column vector of matrix $\mathbf{U}^{(1)}$.

Then, a similarity measure of the super sentence $\mathbf{ss}$ and the sentences represented in the term-sentence-document tensor $\mathcal{X}$ is defined for a sentence s in document i as

$$w_{i,s}^{\text{SS}} = \sqrt{\frac{\sum_{\forall \text{ term } t: t \in \text{document } i} (\|\mathbf{ss}_t\| - \mathcal{X}_{t,s,i})^2}{|\{\forall \text{ term } t : t \in \text{document } i\}|}} \qquad (9)$$

The lower the weight $w_{i,s}^{\text{SS}}$ of a sentence in a document, the more important it is i.e, less the distance of a sentence with the super sentence, more is the similarity. Therefore, an ascending ordering lists the sentences in the order of good candidates for summarization.

D. Data

g) Source Data: We have experimented on several CST data sets, but presented here the results for CST dataset (milan9) [17]. This is a collection of nine coherent single documents related to a Milan plane crash.

h) Preprocessing of documents for creating Tensor: We used stopword filtering and stemming to be the basic preprocessing step creating the tensor from documents.
Stopword filtering is a common technique used to counter the obvious fact that many of the words contained in the document do not contribute particularly to the description of the document content. For instance, words like "the", "is", and "and" contribute very little description, and in many cases they do in fact instead add noise.
Stemming[2] is the process for reducing inflected (or sometimes derived) words to their stem, base or root form generally a written word form.

We collected terms from each document. Then we processed the term list by removing the stop words, and then by stemming them using Porter Stemmer [18]. This is the basic preprocessing phase.

The nine documents have been parsed, tokenized, cleaned, and stemmed. The term list is generated from the whole collection, we created the tensor $\mathcal{X}$ as discussed in Section II. Then by keeping the 15% of the singular values during HOSVD, the TTI has been generated. We have calculated the measures, and defined the sentence significance ranks for each document using both the methods of summarization we discussed.

IV. Experiment

In this section we present the evaluation of the summaries produced by our approach with other standard summarizers. Qualitative analysis of the summaries generated by our method is done by comparing them with human generated summaries and the summaries generated by standard summarizers using ROUGE [19] package.

A. Generation of summaries by available summarizers

i) Open Text Summarizer (OTS): The Open Text Summarizer (OTS)[3] is an open source tool for summarizing texts. The program decides which sentences are important and which are not. It ships with Ubuntu, Fedora and other linux distributions. OTS supports many (25+) languages which are configured in via XML files.

j) MEAD: MEAD[4] [20] is basically a multi-document summarizer. We have used this for summarizing single documents using their demo version.

B. Generation of summaries by Tensor Term Importance (TTI) methods

Our TTI method initially scores the sentences in a document according to their importance. We then arrange the sentences according to their descending value of their scores and extract 30 percent [21] of the sentences (as it is considered to be as good as summary) from the document.

[2] http://en.wikipedia.org/wiki/Stemming
[3] http://libots.sourceforge.net/
[4] http://tangra.si.umich.edu/clair/md/demo.cgi

k) TTI summarization model based on important keywords (T_ImpWords): In document summarization using important terms of TTI, we rank the sentences based on their importance. We have explained this in out previous section. Our assumption lies in the fact that the sentences having more of these important terms are important in the document. So for summarization purpose, we extracted the top ranked 30% of these sentences from each documents.

l) TTI summarization model based on super sentence: resultant of U and S (T_SupSent): We consider a super sentence, which consists of all the words in the document collection. We rank the sentences by giving a score based on the distance measure of a sentence to the super sentence. The less the distance, the more the sentence is similar to the super sentence, which means the sentence is more important. In this case, we extracted the top 30% of the ranked sentences for summarization.

C. Generation of summaries by Human assessors

For evaluation of our method, we involved two human assessors $Human_{RJ}$ and $Human_{BP}$. We gave each of them the sets of documents. We asked them to rank 30% [21] of the important sentences as they read through the text. We then collected those sentences, and then formed extractive summaries maintaining the ranks assigned by them to the sentences. We then used these summaries as a benchmark to analyze qualitatively TTI generated summaries and other automated summarizers like MEAD and OTS.

D. Evaluation

m) ROUGE evaluation for summaries:: ROUGE [19] stands for Recall-Oriented Understudy for Gisting Evaluation. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. The measures count the number of over- lapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries created by humans.

For this experiment, we used two human assessors $Human_{RJ}$ and $Human_{BP}$ and two open source summarizers MEAD and OTS for qualitative analysis of TTI approaches.

In this experiment, we present the result with ROUGE-1 (n-gram approach, with n=1) at 95% confidence level. We enforced the length of the summaries to 100 words. We present only the recall values of the evaluation for each documents. Tables I and II present the average ROUGE-1 recall, precision and F-measure of the automated summarizers and human assessors for all documents.

n) Recall measure: We have evaluated the summaries generated by our both methods with two human assessors. We also evaluated MEAD and OTS with these human assessors and then compared the ROUGE-1 recall scores with our methods. Figure 3 shows the recall comparison of all the automated methods with $Human_{RJ}$.

In this case, we find that the recall curve for our TTI methods are above MEAD and OTS when compared with the summary

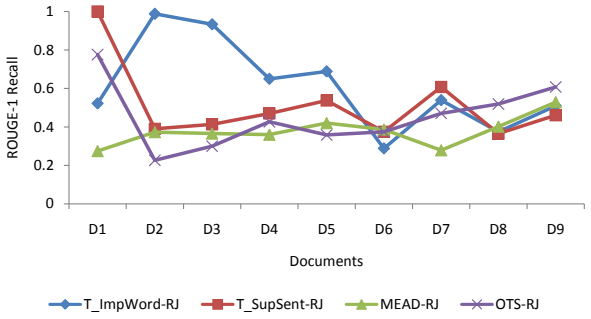

Fig. 3. ROUGE-1 Recall for $Human_{RJ}$ with different automated summarizers

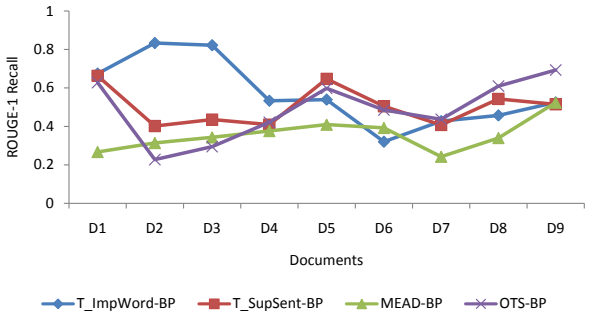

Fig. 4. ROUGE-1 Recall for $Human_{BP}$ with different automated summarizers

generated by $Human_{RJ}$ expect for the last three documents where OTS' performance is better than the others. The average result for *T_ImpWords* is better than *T_SupSent*. With these recall values, we can see that TTI important keyword based summary works well for single document having the reflection of topic words from the whole collection.

Figure 4 illustrates the ROUGE recall results for the summary comparison of all the methods with $Human_{BP}$. For the first four documents *T_ImpWords*' performance are very similar to $Human_{BP}$ than any other summarizers. Then is *T_SupSent*. But from document 5 onwards, *T_SupSent* showed more similarity with $Human_{BP}$ than *T_ImpWords*. OTS showed higher similarity with human for the last two documents. In both the figures, MEAD showed least similarity with humans unlike TTI methods.

Table I depicts that TTI based models are more similar

TABLE I

AVERAGE RECALL, PRECISION AND F-MEASURE OF ROUGE-1 SCORE FOR ALL DOCUMENTS WITH $Human_{RJ}$

Methods	Avg. R	Avg. P	Avg. F
T_ImpWords	0.61	0.54	0.57
T_SupSent	0.51	0.53	0.52
MEAD	0.38	0.39	0.38
OTS	0.45	0.50	0.47

TABLE II

AVERAGE RECALL, PRECISION AND F-MEASURE OF ROUGE-1 SCORE FOR ALL DOCUMENTS WITH $Human_{BP}$

Methods	Avg. R	Avg. P	Avg. F
T_ImpWords	0.56	0.57	0.56
T_SupSent	0.50	0.59	0.53
MEAD	0.36	0.40	0.37
OTS	0.49	0.60	0.53

to human assessed summaries than other two automated summarizers like MEAD and OTS. Similar kind of results are seen in table II, where *T_ImpWords* scores are more similar to $Human_{BP}$ than others. Then is *T_SupSent*. OTS performance is better than MEAD, but not better than TTI based methods.

Through these illustrations we qualitatively analyzed the summaries generated by TTI approaches. The ROUGE scores show that they are qualitatively similar to human than other automated summarizers and we can use these models as good topic oriented summaries.

V. CONCLUSION

In this work, we present two summarization models using Tensor Term Indexing model which uses higher order singular value decomposition for keyword extraction from coherent documents. This model can extract significant terms from a collection of coherent documents. This specifically extracts the topic words which which have been used for ranking sentences for summarization purposes. Our first model is document summarization using important terms of TTI, and the second one is using the super sentence of TTI. The performance of both these models are evaluated with the ROUGE evaluation package. The results show that our former model resembles human assessors more than latter. This shows that our model is qualitatively good for generating topic oriented summaries from a collection of coherent documents unlike the other automated summarizers like MEAD or OTS, which only considers single document at a time for summarization.

VI. ACKNOWLEDGEMENT

The research was supported by HUNOROB project (HU0045), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian National Development Agency.

REFERENCES

[1] H. Luhn, "The automatic creation of literature abstracts," *Advances in Automatic Text Summarization. MIT Press, Cambridge, MA*, pp. 58–63, 1956.

[2] H. P. Edmundson, "New methods in automatic extracting," *J. ACM*, vol. 16, no. 2, pp. 264–285, 1969.

[3] G. Salton, *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

[4] C. Lin and E. Hovy, "Identifying topics by position," in *Proceedings of the fifth conference on Applied natural language processing*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1997, pp. 283–290.

[5] H. Jing and K. R. Mckeown, "Cut and paste based text summarization," in *In Proceedings of the 1st North American Chapter of the Association for Computational Linguistics*, 2000, pp. 178–185.

[6] H. Jing, "Sentence reduction for automatic text summarization," in *In Proceedings of the 6th Applied Natural Language Processing Conference (ANLP00*, 2000, pp. 310–315.

[7] E. D. Liddy, "The discourse-level structure of empirical abstracts: an exploratory study," *Inf. Process. Manage.*, vol. 27, no. 1, pp. 55–81, 1991.

[8] B. Endres-Niggemeyer, E. Maier, and A. Sigel, "How to implement a naturalistic model of abstracting: Four core working steps of an expert abstractor," *Information Processing and Management*, vol. 31, no. 5, pp. 631–674, 1995.

[9] E. Cremmins, "The Art of Abstracting." 1982.

[10] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM New York, NY, USA, 2001, pp. 19–25.

[11] J. Yeh, H. Ke, and W. Yang, "Chinese Text Summarization Using a Trainable Summarizer and Latent Semantic Analysis," *LECTURE NOTES IN COMPUTER SCIENCE*, pp. 76–87, 2002.

[12] R. Mihalcea, "Graph-based ranking algorithms for sentence extraction, applied to text summarization," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Lingusitics (ACL 2004)(companion volume)*, 2004.

[13] ——, "Language independent extractive summarization," in *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*. Association for Computational Linguistics Morristown, NJ, USA, 2005, pp. 49–52.

[14] G. Salton, A. Singhal, M. Mitra, and C. Buckley, "Automatic text structuring and summarization," *Information Processing and Management*, vol. 33, no. 2, pp. 193–207, 1997.

[15] T. Landauer, P. Foltz, and D. Laham, "An Introduction to Latent Semantic Analysis," *DISCOURSE PROCESSES*, vol. 25, pp. 259–284, 1998.

[16] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[17] D. Radev, J. Otterbacher, and Z. Zhang, "Cst bank: A corpus for the study of cross-document structural relationships," in *Proc. of LREC 2004*, 2004.

[18] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 313–316, 1997. [Online]. Available: http://portal.acm.org/citation.cfm?id=275705

[19] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," M.-F. Moens and S. Szpakowicz, Eds. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81.

[20] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Çelebi, E. Drabek, W. Lam, D. Liu, H. Qi, H. Saggion *et al.*, "The MEAD Multidocument Summarizer," 2003.

[21] H. Dalianis, "SweSum-A Text Summarizer for Swedish http://www. dsv. su. se/% 7Ehercules/papers," *Textsumsummary. html*, 2000.