Significant term extraction by Higher Order SVD

Sukanya Manna[1], Zoltán Petres[1,2], and Tom Gedeon[1]
[1]Department 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, we present a novel method for term importance, called Tensor Term Indexing (TTI). This extracts significant terms from a document as well as a coherent collection of document set. The basic idea of this approach is to represent the whole document collection in a Term-Sentence-Document tensor and employs higher-order singular value decomposition (HOSVD) for important term extraction. TTI uses the lower rank approximation technique to reduce noise by eliminating anecdotal terms, to mitigate synonymy by merging 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. Our evaluation shows that that TTI model can extract significant terms relevant to a topic from a small number of documents which Term Frequency and Inverse Document Frequency (tfidf) cannot.**

I. Introduction

In this paper we describe a new approach, the Tensor Term Indexing (TTI) model, to extract the significant terms from a set of coherent documents. It is domain independent and specially applicable to some user specific applications like processing documents of legal case, intelligence related report analysis, summarization [1], and question answering system [2], which require detailed analysis of the texts. This model employs both depth as well as the width characteristics of terms. The width and depth characteristics of a term distribution refer respectively to its distribution within the whole document collection (the number of documents containing the term) and its distribution within the documents containing the term (the number of the terms in these documents). If we consider a sentence containing a term rather than a document as a basic retrieval object, the impact of the depth feature of the term on its significance can be recognized. Here, the terms which are both significant to a single document as well as that whole collection are extracted.

Determination of term importance plays a very important role in achieving high quality indexing. In addition, it is also the basis of automatic classification, automatic indexing, automatic abstracting, search feedback technique and a similarity measure [3], [4], [5], [6]. A wide variety of approaches have been addressed in weighting term importance. They range from the applicable to the theoretical, from the simple to the sophisticated. Some employ statistical theories to calculate term significance [7], some employ artificial neural networks [8], some integrate the latent semantic technique in indexing [9], some apply probability theory to solve the same problem [10], [11], [12], and some just use a simple term frequency method [10], [13]. The pioneer methods are basically [7], [14], [15] which extract the significant terms from a large corpus of data. These methods are mainly concerned with the width characteristics of the terms not the depth. Zhang et al., proposed [16] a term significance model which employs both depth as well as the width characteristics of terms from a collection of documents in a database which might have documents referring to a similar topic as well.

The vector based information retrieval model identifies relevant documents by comparing query terms with terms from a document corpus. This is done by assigning the highest weights to the ones with most discriminative power [14]. Inverse Document Frequency is the most common retrieval model which considers the distribution of terms between documents. There are also modifications of the above concept into inverse sentence frequency and inverse term frequency. Inverse sentence frequency similarly reflects the distribution of terms between sentences and inverse term frequency likewise in sentences or phrases [17]. But for both of these methods the depth and width characteristics are analyzed once at a time.

The concept of singular value decomposition (SVD) has already been applied in Latent Semantic Analysis (LSA) [18] or Latent Semantic Indexing (LSI) [19]. Latent semantic analysis (LSA) is a technique in natural language processing, in particular in vectorial semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. This method involves a huge corpus of data and more query oriented. This concept has been successfully used in search engines.

The previous models are more suitable for generic purposes having a huge corpus of data. Some need training sets, which made these methods unsuitable for analyzing small collection of coherent documents or single document.

The TTI model represents the document in a tensor model form, and using Higher order SVD, the higher dimensional extension of traditional matrix SVD as an underlying mathematical tool to rank the terms in a document set. Unlike LSA, the purpose of our model, TTI is to extract the significant terms from a document using the information of the whole document collection. This is domain independent, has higher specificity, considers both depth and width characteristics of the terms, and computes efficiently in the coherent document collection. The terms extracted using TTI are the keywords that can be used for topic identification, and can also be used to obtain the summaries of the document, or subjective analysis of the documents.

978-1-4244-3802-0/09/$25.00 ©2009 IEEE

The paper is organized as follows: Section II introduce the preliminary modeling and mathematical tools used later. The detailed description of TTI model is given in Section III. An elaborated evaluation is discussed in Section IV, and finally Section V concludes the paper.

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.

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.

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

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 represents 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 is given. The tensor notation, and terminology is based on Lathauwer's work detailed in [20].

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

$$\mathbf{F} = \mathbf{U}_{(1)} \cdot \mathbf{S} \cdot \mathbf{V}_{(2)}^T = \mathbf{S} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{V}_{(2)} = \mathbf{S} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{U}_{(2)} = \mathbf{S} \overset{2}{\underset{n=1}{\otimes}} \mathbf{U}_{(n)}, \tag{2}$$

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:

$$\mathbf{S} = \operatorname{diag}\left(\sigma_1, \sigma_2, \dots, \sigma_{\min(I_1, I_2)}\right), \tag{3}$$

   b) ordering:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(I_1, I_2)} \geq 0. \tag{4}$$

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) 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

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{U}_{(2)} \times_3 \cdots \times_N \mathbf{U}_{(N)} = \mathcal{S} \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_{(n)}, \tag{5}$$

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 $\alpha \neq \beta$:

$$\langle \mathcal{S}_{i_n=\alpha}, \mathcal{S}_{i_n=\beta} \rangle = 0, \text{ when } \alpha \neq \beta, \tag{6}$$

   b) ordering:

$$\|\mathcal{S}_{i_n=1}\| \geq \|\mathcal{S}_{i_n=2}\| \geq \cdots \geq \|\mathcal{S}_{i_n=I_n}\| \geq 0, \tag{7}$$

   for all possible values of n.

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 1.

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.

e) 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

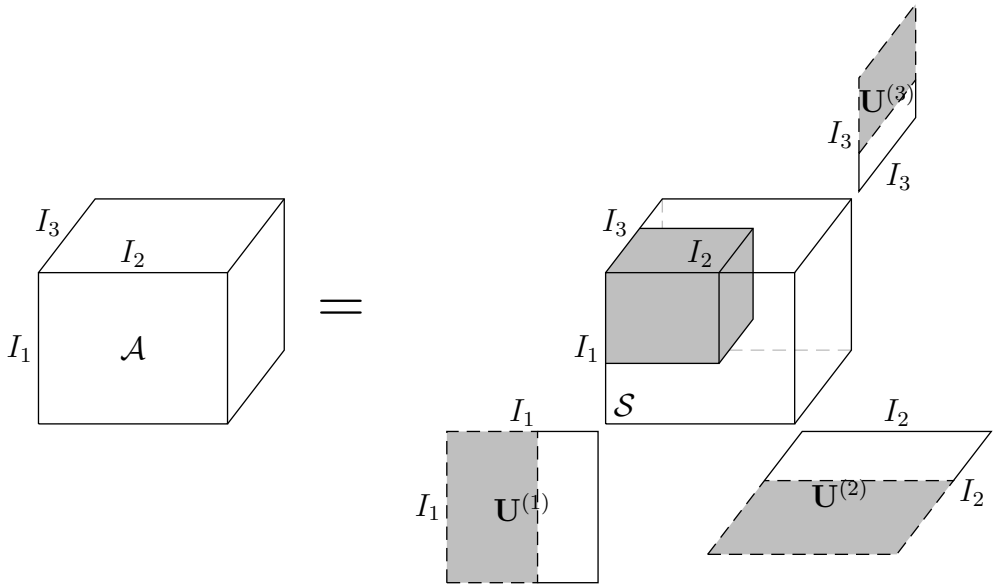

Fig. 1. Visualization of the HOSVD for a third-order tensor

corresponding elements of singular values $\sigma^{(n)}_{I'_n+1}, \sigma^{(n)}_{I'_n+2}, \ldots \sigma^{(n)}_{R_n}$ of tensor $\mathcal{S}$ to zero, for a given $I'_n < R_n$. In this case

$$\gamma = \|\mathcal{A} - \hat{\mathcal{A}}\|^2 \leq \sum_{i_1=I'_{1+1}}^{R_1} \left(\sigma^{(1)}_{i_1}\right)^2 + \sum_{i_2=I'_{2+1}}^{R_2} \left(\sigma^{(2)}_{i_2}\right)^2 + \cdots + \sum_{i_N=I'_{N+1}}^{R_N} \left(\sigma^{(N)}_{i_N}\right)^2. \tag{8}$$

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.

Tensor Term Indexing presents the document in Term-Sentence-Document Tensor format defined in the Preliminaries. 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.

f) 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 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:

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

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:

$$\mathcal{X}_{\text{TTI}} = \mathcal{S}_k \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_{(n)}^k, \tag{9}$$

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.

g) Document level significant term extraction using TTI: TTI is an efficient tool to rank the terms in a document and extract the most significant ones to give a summary. We define the following measure for a term j in document i:

$$w_{i,j} = |\{\forall s, x_{i,s,j} : x_{i,s,j} > 0\}| \sum_{s=1\ldots S} x_{i,s,j}, \tag{10}$$

where $x_{i,s,j}$ is an item of Tensor Term Index $\mathcal{X}_{\text{TTI}}$, S is the number of sentences in the longest document, and $|\{\forall s, x_{i,s,j} : x_{i,s,j} > 0\}|$ gives the number of sentences, where the term has a value higher than zero.

The higher the weight $w_{i,j}$ of a term in a document, the more important it is. Therefore, a descending ordering lists the terms in the order of importance.

h) Term extraction at document collection level using TTI: The significant term ranking can be extended to the whole document collection level, and the significant keywords of the coherent texts can be extracted. The weighting (10) can be modified to the global weighting of term i as

$$w_i = |\{\forall s, \forall d, x_{i,s,d} : x_{i,s,d} > 0\}| \sum_{s=1\ldots S, d=1\ldots D} x_{i,s,d}, \tag{11}$$

where $x_{i,s,d}$ is an item of Tensor Term Index $\mathcal{X}_{\text{TTI}}$, S is the number of sentences in the longest document, D is the number of documents, and $|\{\forall s, \forall d, x_{i,s,d} : x_{i,s,d} > 0\}|$ gives the number of sentences in all documents, where the term has a value higher than zero. The descend ordering of the weight list defines the important terms is priority order in the document collection.

IV. Experiment

In this section we present evaluation of our method by comparing it at different levels with other term extraction models. Our tensor based term significance model is capable of extracting terms both from the single documents as well as from the whole document collection maintaining its relevancy.

A. Term significance models used for evaluation

i) Term Frequency Inverse Document Frequency (tfidf): Tfidf is one of the important term significance model proposed by Salton [21] which is term frequency (tf) x inverse document frequency (idf), where tf is the number of times a term appears in a document, and idf reflects the distribution of terms within the corpus. It is represented as,

$$idf(t_i) = log(N) - log(n_i) + 1 \tag{12}$$

N is the total number of documents in a document collection; n_i is the number of documents that contain at least one occurrence of the term t_i; and t_i is a term, which is typically stemmed.

Ideally, the system should assign the highest weights to terms with the most discriminative power. One component of the corpus weight is the language model used. The most common language model is the Inverse Document Frequency (idf), which considers the distribution of terms between documents.

j) Term Frequency Inverse Sentence Frequency (tfisf): Term frequency inverse sentence frequency [17] is the sentence level modification of the commonly used corpus weighting scheme term frequency inverse document frequency (tfidf). Tf is term frequency here. So, here isf similarly considers the distribution of terms between sentences of a document. Thus it is represented by

$$isf(t_i) = log(N) - log(n_i) + 1 \tag{13}$$

where N is the total number of sentences in a document; n_i is the number of sentences that contain at least one occurrence of the term t_i; and t_i is a term, which is typically stemmed. So, the weight of each term is determined by term frequency (tf) x inverse sentence frequency (isf). Ideally, the system should assign the highest weights to terms with the most discriminative power. One component of the corpus weight is the language model used.

TABLE I
TOP TEN TERMS EXTRACTED AT DOCUMENT COLLECTION LEVEL BY TTI AND PERCENTAGE MATCH OF TOP 30% WORDS EXTRACTED TTI AND TFIDF

Docs	TTI	% Match
All Docs	build, plane, milan, crash, com, news, skyscraper, report office, sai	49.0

TABLE II
TOP TEN TERMS EXTRACTED AT SINGLE DOCUMENT LEVEL BY TTI AND PERCENTAGE MATCH OF TOP 30% WORDS EXTRACTED TTI AND TFISF

Doc	TTI	% Match
D1	build, plane, cnn, com, police, work, crash, floor, people, milan central	82
D2	plane, build, crash, milan, office, abcnews, pirelli, skyscraper, local, floor	79
D3	report, build, milan, immediate, crash, fear, detail, plane, set, city	96
D4	build, com, plane, work, told, cnn, world, pirelli, floor, central	84
D5	build, milan, crash, skyscraper, plane, police, italy, rai, pirelli, pilot	76
D6	build, crash, plane, office, milan, report, abcnews, com, news, pirelli	95
D7	crash, milan, com, build, skyscraper, fox, news, work, inform, press	85
D8	build, plane, air, report, news, office, com, crash, milan, abcnews	88
D9	sai, build, milan, crash, plan, news, man, plane, com skyscraper	88

B. Evaluation

k) Data Set and term selection: We used here CST dataset (milan9) [22]. This is a collection of nine coherent single documents related to a Milan plane crash. 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 [23]. 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 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 term significance for each document and for the whole document collection as well.

l) Document Collection level: In the document level, we consider the whole set of documents for keyword comparison. We have used term frequency and inverse document frequency (tfidf) [21] and our tensor based model to extract significant terms from the whole data set. We have seen that, or model, TTI extracts the relevant terms which are common to all the document collection.

build, plane, milan, crash, com, new, skyscraper, report, offic, sai, pirelli, floor, italian, polic, attack are few top ranked terms we found from the whole document collection using our TTI method. We found a very contrasting result for tfidf method. The terms like *fox, abcnew, foxnew, pilot, local, cnn, inform, control, told, offici, taken, scene, qaeda, detail, today* are the top ranked terms extracted by tfidf, which are not the significant terms as far as the documents context is concerned. The terms shown here by our method are place at the lowest rank in the tfidf, which shows that according to their method they are less important. As the documents of this data set cover a same topic, it tends to decrease the discriminative capacities of tfidf. This contradicts the contextual idea. Table I shows the percentage match of the top 30% words extracted by TTI model and tfidf [24].

Topic words which exists in all the documents are considered to be irrelevant by tfidf. Unlikely it is very essential when we deal with small collection of coherent documents, where the topic words are important for context identification. Besides this, these words are linking the documents to each other in terms of semantic relation. If a term significance model omits topic words, thus for applications like intelligent analysis of documents, summarization, question answering, the results will not at all be relevant and justifiable. As the results show, our model fits best in this regard.

If we observe the pattern of occurrence of words in the single documents (Table II), it is clearly seen that the words like *build, plane, crash, milan, skyscraper* etc are the words which exist almost in all the documents. These words are again extracted by our method from the whole document collection which is significant enough to identify a contextual concept.

m) Single Document Level: In this part we present the results based on the single documents. We use term frequency and inverse sentence frequency (tfisf) as another model, a similar modification of tfidf at sentence level. We present here (Table II) top ten [1,2] (as most people prefer 10 to be a popular number for presentation of any significant results) significant terms produced by our method. We extracted top 30% words ranked by TTI as well as tfisf for each documents. Then we have found out the percentage of words common in both the lists, which we have shown in Table II along with the words extracted.

In this case, we find that TTI as well as tfisf works almost in the similar way. Though tfisf is the modification of tfidf, it seems to work reasonably good at sentence level for single documents, which we can see by the percentage match between two word lists. When sentences are considered, they are fairly different from each other, which helped tfisf to work better at this level maintaining a reasonable discrimination among words. Our model extracts very relevant terms here as well. Both at the document collection level, and at single document level, the results are consistent and topic relevant.

[1] http://tempodrive.com/index.php?option=com_content&task=view&id=33&Itemid=63

[2] http://searchenginewatch.com/reports

V. Conclusion

In this work, we present a Tensor Term Indexing model which uses higher order singular value decomposition for keyword extraction from coherent documents. This model can extract significant terms both at document collection level as well as single document level. This specifically extracts the topic words which are actually useful for many other applications like summarization, topic identification, intelligent analysis of documents and so on.

We have shown here that tfidf is not good to extract terms in this case. It is incapable of identifying topic words at the whole document level. On the contrary, our model, TTI, works well in this regard.

Since this model is purely mathematical, so we intend to create a multilingual platform of keyword extraction, as we aim to find the semantic relations without really concerned with the linguistic part. We start with simple term frequency approach, where neither previous knowledge of the document is required not the language of the document.

VI. Acknowledgement

Dr. Petres was supported by Group of Eight European Fellowship for participating in this research at The Australian National University. The research was supported also by HUNOROB project (HU0045), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian Focal Point.

References

[1] Z. Zhang, S. Blair-Goldensohn, and D. R. Radev, "Towards cst-enhanced summarization," in *Eighteenth nat. conf. on AI*. Menlo Park, CA, USA: AAAI, 2002, pp. 439–445.

[2] B. Katz and et al, "Start, natural language question answering system," 1993.

[3] F. Debole and F. Sebastiani, "Supervised term weighting for automated text categorization," in *SAC '03: Proc. of the 2003 ACM symp. on Applied computing*. New York, NY, USA: ACM, 2003, pp. 784–788.

[4] Y.-S. Lai and C.-H. Wu, "Meaningful term extraction and discriminative term selection in text categorization via unknown-word methodology," *ACM Trans. on Asian Language Info. Proces. (TALIP)*, vol. 1, no. 1, pp. 34–64, 2002.

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

[6] B. UMINO, "Some Principles of Weighting Methods Based on Word Frequencies for Automatic Indexing," *Lib. and info. sc.*, vol. 26, pp. 67–88, 1988.

[7] T. Noreault, M. McGill, and M. Koll, "A performance evaluation of similarity measures, document term weighting schemes and representations in a Boolean environment," in *Proc. of the 3rd annual ACM conf. on Research and development in IR*. Butterworth & Co. Kent, UK, UK, 1980, pp. 57–76.

[8] Z. Boger, T. Kuflik, P. Shoval, and B. Shapira, "Automatic keyword identification by artificial neural networks compared to manual identification by users of filtering systems," *Info. Process. and Mgmt.*, vol. 37, no. 2, pp. 187–198, 2001.

[9] M. Gordon and S. Dumais, "Using latent semantic indexing for literature based discovery," *J. of the American Society for Info. Sc.*, vol. 49, no. 8, pp. 674–685, 1998.

[10] W. Greiff, "A theory of term weighting based on exploratory data analysis," in *Proc. of the 21st annual int. ACM SIGIR conf. on Research and development in IR*. ACM New York, NY, USA, 1998, pp. 11–19.

[11] M. Melucci, "Passage retrieval: A probabilistic technique," *Info. Process. and Mgmt.*, vol. 34, no. 1, pp. 43–68, 1998.

[12] J. Ponte and W. Croft, "A language modeling approach to information retrieval," in *Proc. of the 21st annual int. ACM SIGIR conf. on Research and development in IR*. ACM New York, NY, USA, 1998, pp. 275–281.

[13] G. Salton and C. Yang, "On the Specification of Term Values in Automatic Indexing," 1973.

[14] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," Ithaca, NY, USA, Tech. Rep., 1987.

[15] H. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM J. of Research and Development*, vol. 1, no. 4, pp. 309–317, 1957.

[16] J. Zhang and T. N. Nguyen, "A new term significance weighting approach," *J. Intell. Info. Syst.*, vol. 24, no. 1, pp. 61–85, 2005.

[17] C. Blake, "A comparison of document, sentence, and term event spaces," in *ACL-44: Proc. of the 21st Int. Conf. on Comput. Lingu. and the 44th annual meeting of the ACL*. Morristown, NJ, USA: ACL, 2006, pp. 601–608.

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

[19] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[20] 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.

[21] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Info. Process. and Mgmt.*, vol. 24, no. 5, pp. 513–523, 1988.

[22] 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.

[23] 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

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