International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 3 No. 3 July 2013, pp. 701-713

http://www.issr-journals.org/ijias/

Effect of Near-orthogonality on Random Indexing Based Extractive Text Summarization

Niladri Chatterjee[1] and Pramod K. Sahoo[2]

[1]Department of Mathematics,
Indian Institute of Technology Delhi,
Hauz Khas, New Delhi 110016, India

[2]Institute for Systems Studies and Analyses,
Defence Research and Development Organisation,
Metcalfe House Complex, Delhi 110054, India

ABSTRACT: Application of Random Indexing (RI) to extractive text summarization has already been proposed in literature. RI is an approximating technique to deal with high-dimensionality problem of Word Space Models (WSMs). However, the distinguishing feature of RI from other WSMs (e.g. Latent Semantic Analysis (LSA)) is the near-orthogonality of the word vectors (index vectors). The near-orthogonality property of the index vectors helps in reducing the dimension of the underlying Word Space. The present work focuses on studying in detail the near-orthogonality property of random index vectors, and its effect on extractive text summarization. A probabilistic definition of near-orthogonality of RI-based Word Space is presented, and a thorough discussion on the subject is conducted in this paper. Our experiments on DUC 2002 data show that while quality of summaries produced by RI with Euclidean distance measure is almost invariant to near-orthogonality of the underlying Word Space; the quality of summaries produced by RI with cosine dissimilarity measure is strongly affected by near-orthogonality. Also, it is found that RI with Euclidean distance measure performs much better than many LSA-based summarization techniques. This improved performance of RI-based summarizer over LSA-based summarizer is significant because RI is computationally inexpensive as compared to LSA which uses Singular Value Decomposition (SVD) - a computationally complex algebraic technique for dimension reduction of the underlying Word Space.

KEYWORDS: Word Space, Random Indexing, index vector, context vector, near-orthogonal, PageRank.

1 INTRODUCTION

Random Indexing (RI) along with PageRank based algorithm for extractive text summarization has already been proposed in literature by Chatterjee and Mohan [1]. The overall scheme proposed therein is based on the following steps:

- Representation of words using fixed dimension (d, say) ternary index vectors, containing 0, +1 and -1;
- Aggregation of the word vectors first into context vectors, and then from there to sentence vectors;
- Construction of a document graph from the sentence vectors by establishing links between nodes representing sentences;
- Application of PageRank algorithm to rank the sentences according to their importance;
- Finally, selection of the desired number of sentences to form the extractive summary of the document.

Random Indexing [2], [3] is an approximating technique to deal with high-dimensionality problem of Word Space Models (WSMs) [3], [4]. The advantage of this approach is that neither it requires any language-specific resources (e.g. on-line dictionary, thesaurus), nor does it depend on any informed heuristics of the underlying language. A major difference between RI with the most commonly used WSM based approach viz. Latent Semantic Analysis (LSA) [5], [6] is that LSA uses orthogonal unary vectors to represent words of a document, and builds a semantic space over it. On the other hand, in an RI-

Corresponding Author: Niladri Chatterjee (niladri.iitd@gmail.com)

based Word Space the index vectors are not orthogonal. As discussed in section 4, there is some probability of two distinct index vectors being non-orthogonal to each other. This property is called *near-orthogonality* of the index vectors. This near-orthogonality property of index vectors is the key to the RI-methodology as it helps in reducing the dimension of the Word Space before building the context information of a text passage. A study of near-orthogonality is therefore considered primary before advocating RI as a viable alternative to other WSM based approaches for extractive summarization.

A preliminary study in [7] shows that near-orthogonality of index vectors has an effect on the quality of summary produced by RI-based summarization schemes. The primary focus of the present work is to conduct an in-depth study of the effect of near-orthogonality of index vectors on the quality of summaries produced by RI with PageRank based scheme which we term as Random Indexing based Summarization (RISUM). In this study we experimented RISUM with both angular and linear proximity measures, used a larger dataset than [7], and utilized ROUGE [8] metrics to measure the quality of the summaries.

The rest of the paper is organized as follows. Section 2 gives a brief description of LSA-based summarization techniques proposed in literature. In section 3 Random Indexing and RISUM approach for extractive summarization has been presented. Section 4 elucidates the concept of near-orthogonality in an RI-based Word Space. Section 5 deals with experimental details, findings on effect of near-orthogonality on the performance of RISUM and comparative performance evaluation of LSA-based summarizers with RISUM. Section 6 presents concluding remarks.

2 LSA-BASED TEXT SUMMARIZATION

LSA is a Word Space implementation technique built upon the distributional information of words over a text passage. It uses Singular Value Decomposition (SVD); an algebraic dimension reduction technique, to extract the contextual information between the text units (words, sentences, or paragraphs) of the passage. In this technique each distinct word in a document is represented by a distinct unary vector. The dimension of each word vector is equal to the number of distinct words in the document. Each sentence in the document is represented by a sentence vector, which is constructed by taking the weighted algebraic sum of the word vectors. Suppose a document consists of m distinct words $w_1, w_2, \cdots, w_m$; whose word vectors are m-dimensional unary vectors $\vec{w}_1 = [1\ 0\ 0\ \cdots\ 0]^T$, $\vec{w}_2 = [0\ 1\ 0\ \cdots\ 0]^T$,..., $\vec{w}_m = [0\ 0\ 0\ \cdots\ 1]^T$ respectively. If the document consists of n number of sentences $s_1, s_2, \cdots, s_n$ and the weight assigned to the word w_i in sentence s_j is f_{ij}, then the sentence vector of sentence s_j is given by $\vec{s}_j = \sum_{i=1}^{m} f_{ij}\vec{w}_i$. In this way the document can be represented as a words-by-sentences matrix $S = [\vec{s}_1\ \vec{s}_2\ \cdots\ \vec{s}_n]$ of size $m \times n$. Then SVD is performed on matrix S, so that it is factored into a product of three matrices: $S = U\Sigma V^T$; where columns of U are the left singular vectors of S, Σ is a diagonal matrix consisting of singular values of S, and columns of V are the right singular vectors of S. The latent semantic structure of document represented by matrix S can be derived from its factors U, Σ and V [9].

A number of weighting schemes to construct sentence vectors from unary word vectors have been proposed in literature. Ozsoy *et al.* [10] gives a brief description of some of these weighting schemes, which are: (i) word frequency, (ii) binary, (iii) *Tf-Idf* (term (word) frequency-inverse document frequency), (iv) log entropy, (v) root type, and (vi) modified *Tf-Idf*.

Important sentences of a document can be extracted by exploiting the factors U, Σ and V of S. Some of the significant contributions in this regard are Gong and Liu [11], Steinberger and Ježek [12], Murray *et al.* [13] and Ozsoy *et al.* [10], [14]. A comparative study on performance evaluation of these methods were conducted in Ozsoy *et al.* [10], [14] by considering different weighting schemes as mentioned above. Sahlgren [2], [3] criticized LSA for being both computationally expensive and requiring the formation of a full co-occurrence matrix and its decomposition before any similarity computation can be performed. On the other hand, RI allows the incremental building of the semantic space by creating a short index vector for each unique context, and producing the context vector for each word by summing index vectors for each context as one scan through the text.

3 RANDOM INDEXING AND RISUM

This section gives a brief introduction of RI and the summarization scheme RISUM.

3.1 RANDOM INDEXING

RI initially assigns each distinct word in the document a unique and randomly generated vector called the *index vector* of the respective word. These index vectors are sparse, high-dimensional, and ternary. Each index vector of dimension d consists of a large number of '0's and a small number (ε) of '+1's and '-1's with the following probabilities:

$$\begin{cases} +1 \text{ with probability } \dfrac{\varepsilon}{2d} \\ 0 \text{ with probability } \dfrac{d-\varepsilon}{d} \\ -1 \text{ with probability } \dfrac{\varepsilon}{2d} \end{cases} \quad (1)$$

The major advantage of using these index vectors is that they can handle large Word Spaces very efficiently. For example, in LSA vectors of dimension $d = 1000$ can cover a vocabulary of 1000 words only, whereas the same dimension of 1000 with one '+1' and one '-1' can construct 999000 (1000×999) index vectors, and hence can cover a vocabulary of 999000 (~ 10^6) words.

In order to represent the semantics of a document it is assumed that the intended semantics of a word in a document can be found from the context words of the document. This is done by computing the *context vector* for each distinct word. Initially the context vector of each distinct word in the document is initialized to the d-dimensional null vector. Every time a word w_k occurs in the document, the index vectors of the words in its context window are added to its context vector. For example, if a $2 + 2$ sized context window (i.e. a context window spread over two words on either side of the focus word) represented by $[(w_{k-2}w_{k-1})w_k(w_{k+1}w_{k+2})]$ is considered, then the context vector of w_k would be updated as:

$$\mathcal{C}(w_k) := \mathcal{I}(w_{k-2}) + \mathcal{I}(w_{k-1}) + \mathcal{C}(w_k) + \mathcal{I}(w_{k+1}) + \mathcal{I}(w_{k+2}) \quad (2)$$

where $\mathcal{I}(w_k)$ and $\mathcal{C}(w_k)$ respectively are the index vector and context vector of the word w_k. Thus the context vectors of all words in the document are built incrementally by scanning their occurrences in the document.

3.2 RANDOM INDEXING BASED SUMMARIZATION

The major steps of RISUM are as follows:

- Mapping of words into the RI-based Word Space;
- Mapping of sentences into the RI-based Word Space;
- Representing the document as a proximity graph;
- Summary generation.

Subsections 3.2.1 to 3.2.4 discuss these steps in detail. Before being subjected to summarization, a document is pre-processed to identify sentence end-markers and *content words*[1]. The content words of the document are identified by using a list of stop words. We used 'smart_common_words.txt'[2] file which consists of 598 stop words. Then Porter stemming algorithm [15] is used to remove the common morphological and inflectional endings from the words.

3.2.1 MAPPING OF WORDS INTO THE RI-BASED WORD SPACE

Initially each distinct word in the document is assigned a unique index vector of dimension d with '+1's being placed randomly anywhere in the upper half, and the '-1's being placed randomly anywhere in the lower half positions. The context vector for each content word is then constructed by considering a $2 + 2$ sized context window. First, the context vector of each distinct content word in the document is initialized to the d-dimensional null vector. The context of a content word is restricted within the underlying sentence in which the word had occurred. Every time a content word w_k occurs in the document, its context vector $\mathcal{C}(w_k)$ is updated using equation (2). However, to give more weightage to the words nearer to the focus word we have used a weight vector $[0.5 \quad 1 \quad 1 \quad 0.5]$ which assigns weight 1 to the words adjacent to the focus word and weight 0.5 to the words which are at a distance 2.

[1] *Content words are the key words of a sentence. They are the important words that carry the meaning or sense.*

[2] *ROUGE, the evaluation metric used by DUC also uses the same file for listing stop words.*

3.2.2 MAPPING OF SENTENCES INTO THE RI-BASED WORD SPACE

The sentences of the document are mapped into the Word Space by constructing the sentence vectors using equation (3) below:

$$\mathcal{S}(s_j) = \frac{1}{m_j}\sum_{i=1}^{m_j}\left(\mathcal{C}(w_{ij}) - \mathcal{O}\right) \tag{3}$$

where,

- $\mathcal{S}(s_j)$ is the sentence vector of the j-th sentence;
- $\mathcal{C}(w_{ij})$ is the context vector of the i-th content word of j-th sentence;
- m_j is the number of content words in the j-th sentence;
- $\mathcal{O}$ is the central theme of the document computed as arithmetic mean of context vectors of content words of the document.

Subtraction of the mean vector from the context vector in equation (3) reduces the magnitude of the context vectors close in the direction to the central theme of the document, and increases the magnitude of context vectors which are almost in the opposite direction from the central theme. This reduces the influence of the commonly occurring words, such as the auxiliary verbs, articles, on the sentence vector [16].

3.2.3 REPRESENTING THE DOCUMENT AS A PROXIMITY GRAPH

The whole document is represented as a proximity graph, where the nodes of the graph represent the sentences of the document, and weighted edges represent the proximity between sentences. The proximity between any two sentences can be calculated in many different ways. In our implementation we have used two generic schemes: the *angular distance*, and the *linear distance*. In particular, we used (i) Cosine dissimilarity measure as angular distance, and (ii) Euclidean distance to measure the linear distance. Equations (4) and (5) give mathematical formula for the two proximity measures, respectively:

- Cosine dissimilarity measure: $c_{ij} = 1 - \frac{\sum_{k=1}^{d}(s_{ik}\cdot s_{jk})}{\sqrt{\sum_{k=1}^{d} s_{ik}^2}\cdot\sqrt{\sum_{k=1}^{d} s_{jk}^2}}$ (4)

- Euclidean distance measure: $e_{ij} = \sqrt{\sum_{k=1}^{d}\left(s_{ik} - s_{jk}\right)^2}$ (5)

Where $\mathcal{S}(s_q) = [s_{q1} \quad s_{q2} \quad \cdots \quad s_{qd}]$ is the q-th sentence vector.

3.2.4 SUMMARY GENERATION

The weighted PageRank algorithm [17] has been used to get rid of the redundant information in the text by removing the sentences of less importance. If $G = (V, E)$ be an undirected graph with the set of nodes V and set of edges E, then the weighted PageRank of a node v_i denoted by $PR^W(v_i)$ is defined as:

$$PR^W(v_i) = (1-\tau) + \tau \sum_{v_j \in \{v_i : \{v_i, v_j\} \in E\}} \frac{\omega_{ij} PR^W(v_j)}{\sum_{v_k \in \{v_j : \{v_j, v_k\} \in E\}} \omega_{jk}} \tag{6}$$

where ω_{ij} is the weight associated with the undirected edge $\{v_i, v_j\}$ and $\omega_{ij} = \omega_{ji}$ for all i, j. τ is a parameter chosen between 0 and 1 and set to 0.85 as per the recommendation of Brin and Page [18].

Iterative application of the weighted PageRank algorithm on the proximity graph makes the node weights converge. In case of cosine dissimilarity measure, heavily weighted nodes are considered for summary generation; while in case of Euclidean distance measure, light-weight nodes are picked up for the summary.

In our implementation of the summarizer we made two deviations from the one proposed by Chatterjee and Mohan [1].

1. We put restrictions on the placement of '+1's and '-1's in an index vector (see subsection 3.2.1). We found that random positioning of '+1's and '-1's may result in creating two index vectors having '+1' and '-1' occurring at the same coordinate position. This in turn creates erroneous context vectors as '+1' and '-1' cancel each other. To avoid this situation we restrict the placement of '+1's in the upper half positions and '-1's in the lower half

positions of an index vector. This restriction ensures that addition of index vectors while forming a context vector do not end up in producing a null vector; hence the semantics is not misrepresented.

2. Instead of cosine similarity we have used dissimilarity measure. The metric used by them is:

$$C_{ij} = \frac{\sum_{k=1}^{d}(s_{ik} \cdot s_{jk})}{\sqrt{\sum_{k=1}^{d} s_{ik}^2} \cdot \sqrt{\sum_{k=1}^{d} s_{jk}^2}} \quad (7)$$

However, we found that this similarity metric maps the edge weight between any two nodes into the interval [-1, 1]. It is therefore possible that with a mixture of positive and negative edge weights, PageRank value of every node will diverge to either ∞ or $-\infty$. For example, if the weighted PageRank algorithm with $\tau = 0.85$ is applied iteratively on a 3-node undirected graph, where:

- Edge weights are: $\omega_{12} = \omega_{21} = -0.5,\ \omega_{13} = \omega_{31} = 0.9$, and $\omega_{23} = \omega_{32} = 0.7$;
- Initial PageRank of all the three nodes are equal to 0.1;

Then $PR^W(v_1) \to -\infty$, $PR^W(v_2) \to \infty$, and $PR^W(v_2) \to \infty$. This had been an inherent drawback of the earlier scheme. We have not computed theoretically (or estimated experimentally) the probability of occurrence for this divergence. However, in our experiments with the DUC 2002 [19] datasets this divergence had occurred very frequently. This problem has now been rectified by taking the simple linear transformation of the similarity measure: $c_{ij} = 1 - C_{ij}$ (see equations (4) and (7)), which now confines edge weights in the interval [0, 2], and thereby annihilates completely any chance of eventual divergence of the weighted PageRank algorithm.

4 NEAR-ORTHOGONALITY IN A RI-BASED WSM

Let $\mathbb{W}_{RI}$ be a RI-based Word Space consisting of a set of m d-dimensional index vectors $\mathbb{I} = \{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_m\}$. Any two index vectors $\mathcal{I}_j$ and $\mathcal{I}_k$ are said to be orthogonal if their dot product is zero, i.e., $\mathcal{I}_j \cdot \mathcal{I}_k = 0$. This happens when positions of '+1' and '-1's of both the index vectors are different. That means if for any $p \in \{1,2,\cdots,d\}$, the p-th coordinate of $\mathcal{I}_j$ contains either '+1' or '-1', then the p-th coordinate of $\mathcal{I}_k$ must contain a zero, and vice-versa. Since an index vector consists of large number of zeros, there is a high probability that any two index vectors chosen randomly are still orthogonal. However, there is a small probability that the two randomly chosen index vectors are not orthogonal. This probability depends upon the parameters d (the length of the index vector), and ε (the number of '+1's and '-1's in an Index vector) (see equation (1)). Hence we term an RI-based Word Space as near-orthogonal. Near-orthogonality of an RI-based Word Space can be expressed in a probabilistic way in terms of a parameter β; as β-orthogonal, where β is an angle between 0 to $\pi/2$ radian. The following subsection provides a mathematical definition for β-orthogonality with respect to random index vectors.

4.1 β-ORTHOGONALITY IN A RI-BASED WORD SPACE

For $\mathbb{W}_{RI}$ consisting of a set of m index vectors $\mathbb{I} = \{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_m\}$, consider $\mathbb{J} \subseteq \mathbb{I}$. Let $\mathcal{I}_k \in \mathbb{I}/\mathbb{J}$ and $\mathbb{J}_\beta = \left\{\mathcal{I}_j \in \mathbb{J}: \left|Ang(\mathcal{I}_j, \mathcal{I}_k) - \frac{\pi}{2}\right| \leq \left(\frac{\pi}{2} - \beta\right)\right\}$, where β is a predetermined angle between 0 to $\pi/2$ radian, and $Ang(\mathcal{I}_j, \mathcal{I}_k)$ is the angle between the index vectors $\mathcal{I}_j$ and $\mathcal{I}_k$ measured in radian. The subset $\mathbb{J}$ is said be *β-orthogonal* to the index vector $\mathcal{I}_k$ with probability p, where $p = |\mathbb{J}_\beta|/|\mathbb{J}|$.

It may be noted that, the subset $\mathbb{J}$ contains $p \cdot |\mathbb{J}|$ number of index vectors which deviate at most by an angle of $\left(\frac{\pi}{2} - \beta\right)$ radian from being orthogonal to the index vector $\mathcal{I}_k$. These index vectors constitute the subset $\mathbb{J}_\beta$ and lie in the grey colored region as shown in a representative two dimensional plot (see Figure 1). One can also represent the subset $\mathbb{J}_\beta$ as $\mathbb{J}_\beta = \{\mathcal{I}_j \in \mathbb{J}: \beta \leq Ang(\mathcal{I}_j, \mathcal{I}_k) \leq \pi - \beta\}$.

If $\tilde{\mathbb{J}} = \{(\mathcal{I}_j, \mathcal{I}_k): (\mathcal{I}_j, \mathcal{I}_k) \in \mathbb{I} \times \mathbb{I} \text{ and } j < k\}$ and $\tilde{\mathbb{J}}_\beta = \left\{(\mathcal{I}_j, \mathcal{I}_k) \in \tilde{\mathbb{J}}: \left|Ang(\mathcal{I}_j, \mathcal{I}_k) - \frac{\pi}{2}\right| \leq \left(\frac{\pi}{2} - \beta\right)\right\}$, then the RI-based Word Space $\mathbb{W}_{RI}$ is said to be *β-orthogonal* with probability p, where $p = |\tilde{\mathbb{J}}_\beta|/|\tilde{\mathbb{J}}|$.

The Word Space $\mathbb{W}_{RI}$ is close to orthogonality if for large β $(0 < \beta \leq \pi/2)$ the value of p is large. It can be noted that if for $\beta = \pi/2$ the value of p is 1, then the Word Space $\mathbb{W}_{RI}$ is orthogonal. In our discussion, the terms 'orthogonal' and '$\pi/2$-orthogonal' have different meanings. If $\mathbb{W}_{RI}$ is $\pi/2$-orthogonal with $p = 1$, then we call it as orthogonal; otherwise for $p < 1$, we call it as $\pi/2$-orthogonal with probability p.

If in a RI-based Word Space $\mathbb{W}_{RI}$, angle between any two index vectors is one of the angles $\beta_1, \beta_2, \cdots, \beta_{r-1}$ or β_r such that $\beta_1 > \beta_2 > \cdots > \beta_{r-1} > \beta_r$, then $\left|\tilde{\mathbb{J}}_{\beta_t}\right| = \left|\tilde{\mathbb{J}}_{\beta_{t-1}}\right| + N_{\beta_t}$, where $t = 2,3,\cdots,r$ and $N_{\beta_t} = \left|\left\{\left(\mathcal{J}_j, \mathcal{J}_k\right) \in \tilde{\mathbb{J}} : Ang\left(\mathcal{J}_j, \mathcal{J}_k\right) = \beta_t\right\}\right|$.

In order to study the effect of near-orthogonality in a RI-based Word Space we consider following three cases as per equation (1):

Case 1: $\varepsilon = 2$, i.e. index vectors consisting of one '+1' and one '-1';

Case 2: $\varepsilon = 4$, i.e. index vectors consisting of two '+1's and two '-1's;

Case 3: $\varepsilon = 6$, i.e. index vectors consisting of three '+1's and three '-1's.

Subsections 4.1.1 to 4.1.3 discuss these three cases. In this discussion $\mathrm{C}(m, n)$ denotes the combination function: the number of ways of selecting n objects out of m objects. The dimension of index vectors are taken as even numbers, say $d = 2k$. The occurrence of '+1's is restricted in the upper half (positions 1 to k) and occurrence of the '-1's in the lower half (positions $k + 1$ to $2k$) of an index vector. Calculation of different probabilities is based on the above positional restrictions that we have followed in our implementation.

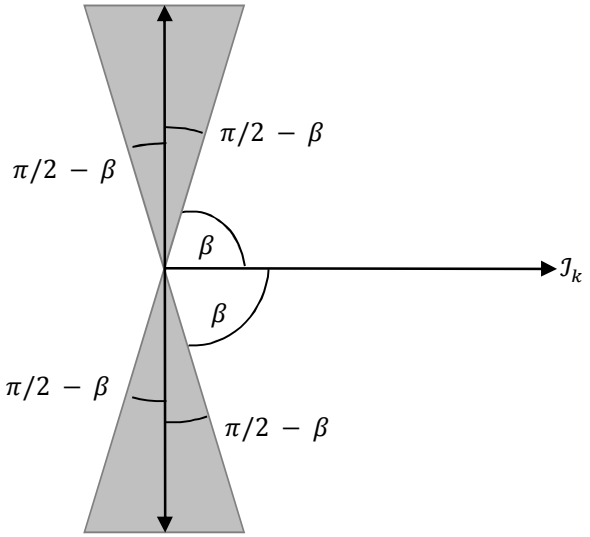

Fig. 1. Near-orthogonality w.r.t. an index vector in two-dimensional plane

4.1.1 CASE-1: INDEX VECTORS CONSISTING OF ONE '+1' AND ONE '-1'

In this case $|\mathbb{I}| = k^2$ and $\left|\tilde{\mathbb{J}}\right| = \mathrm{C}(k^2, 2)$. Any two index vectors are either orthogonal to each other or they make an angle of $\pi/3$ radian between them. Here,

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{2}}\right| = \frac{1}{2}k^2(k-1)^2 \tag{8}$$

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{3}}\right| = \left|\tilde{\mathbb{J}}_{\frac{\pi}{2}}\right| + k^2(k-1) \tag{9}$$

Hence, the RI-based Word Space is $\pi/2$-orthogonal with probability $\left|\tilde{\mathbb{J}}_{\frac{\pi}{2}}\right| / \left|\tilde{\mathbb{J}}\right|$ and $\pi/3$-orthogonal with probability 1.

4.1.2 CASE-2: INDEX VECTORS CONSISTING OF TWO '+1'S AND TWO '-1'S

In this case $|\mathbb{I}| = \left(\mathrm{C}(k, 2)\right)^2$ and $\left|\tilde{\mathbb{J}}\right| = \mathrm{C}\left(\left(\mathrm{C}(k, 2)\right)^2, 2\right)$. Any two index vectors are either orthogonal to each other or they make an angle of $\frac{\pi}{2.383}, \frac{\pi}{3}$ or $\frac{\pi}{4.347}$ radian between them. Here,

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{2}}\right| = \frac{1}{2}\left(\mathrm{C}(k-2, 2)\right)^2\left(\mathrm{C}(k, 2)\right)^2 \tag{10}$$

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{2.383}}\right| = \left|\tilde{\mathbb{J}}_{\frac{\pi}{2}}\right| + 2(k-2)\mathrm{C}(k-2,2)\big(\mathrm{C}(k,2)\big)^2 \tag{11}$$

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{3}}\right| = \left|\tilde{\mathbb{J}}_{\frac{\pi}{2.383}}\right| + (\mathrm{C}(k-2,2) + 2(k-2)^2)\big(\mathrm{C}(k,2)\big)^2 \tag{12}$$

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{4.347}}\right| = \left|\tilde{\mathbb{J}}_{\frac{\pi}{3}}\right| + 2(k-2)\big(\mathrm{C}(k,2)\big)^2 \tag{13}$$

Hence the RI-based Word Space is $\frac{\pi}{2}$-orthogonal with probability $\left|\tilde{\mathbb{J}}_{\frac{\pi}{2}}\right|/|\tilde{\mathbb{J}}|$, $\frac{\pi}{2.383}$-orthogonal with probability $\left|\tilde{\mathbb{J}}_{\frac{\pi}{2.383}}\right|/|\tilde{\mathbb{J}}|$, $\frac{\pi}{3}$-orthogonal with probability $\left|\tilde{\mathbb{J}}_{\frac{\pi}{3}}\right|/|\tilde{\mathbb{J}}|$, and $\frac{\pi}{4.347}$-orthogonal with probability 1.

4.1.3 CASE-3: INDEX VECTORS CONSISTING OF THREE '+1'S AND THREE '-1'S

In this case $|\mathbb{I}| = \big(\mathrm{C}(k,3)\big)^2$ and $|\tilde{\mathbb{J}}| = \mathrm{C}\left(\big(\mathrm{C}(k,3)\big)^2, 2\right)$. Any two index vectors are either orthogonal to each other or they make an angle of $\frac{\pi}{2.239}, \frac{\pi}{2.552}, \frac{\pi}{3}, \frac{\pi}{3.735}$ or $\frac{\pi}{5.364}$ radian between them. Here,

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{2}}\right| = \frac{1}{2}\big(\mathrm{C}(k-3,3)\big)^2\big(\mathrm{C}(k,3)\big)^2 \tag{14}$$

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{2.239}}\right| = \left|\tilde{\mathbb{J}}_{\frac{\pi}{2}}\right| + 3\mathrm{C}(k-3,2)\mathrm{C}(k-3,3)\big(\mathrm{C}(k,3)\big)^2 \tag{15}$$

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{2.552}}\right| = \left|\tilde{\mathbb{J}}_{\frac{\pi}{2.239}}\right| + \frac{1}{2}\Big(6(k-3)\mathrm{C}(k-3,3) + 9\big(\mathrm{C}(k-3,2)\big)^2\Big)\big(\mathrm{C}(k,3)\big)^2 \tag{16}$$

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{3}}\right| = \left|\tilde{\mathbb{J}}_{\frac{\pi}{2.552}}\right| + \big(\mathrm{C}(k-3,3) + 9(k-3)\mathrm{C}(k-3,2)\big)\big(\mathrm{C}(k,3)\big)^2 \tag{17}$$

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{3.735}}\right| = \left|\tilde{\mathbb{J}}_{\frac{\pi}{3}}\right| + \frac{1}{2}(6\mathrm{C}(k-3,2) + 9(k-3)^2)\big(\mathrm{C}(k,3)\big)^2 \tag{18}$$

$$\left|\tilde{\mathbb{J}}_{\frac{\pi}{5.364}}\right| = \left|\tilde{\mathbb{J}}_{\frac{\pi}{3.735}}\right| + 3(k-3)\big(\mathrm{C}(k,3)\big)^2 \tag{19}$$

Hence the RI-based Word Space is $\frac{\pi}{2}$-orthogonal with probability $\left|\tilde{\mathbb{J}}_{\frac{\pi}{2}}\right|/|\tilde{\mathbb{J}}|$, $\frac{\pi}{2.239}$-orthogonal with probability $\left|\tilde{\mathbb{J}}_{\frac{\pi}{2.239}}\right|/|\tilde{\mathbb{J}}|$, $\frac{\pi}{2.552}$-orthogonal with probability $\left|\tilde{\mathbb{J}}_{\frac{\pi}{2.552}}\right|/|\tilde{\mathbb{J}}|$, $\frac{\pi}{3}$-orthogonal with probability $\left|\tilde{\mathbb{J}}_{\frac{\pi}{3}}\right|/|\tilde{\mathbb{J}}|$, $\frac{\pi}{3.735}$-orthogonal with probability $\left|\tilde{\mathbb{J}}_{\frac{\pi}{3.735}}\right|/|\tilde{\mathbb{J}}|$, and $\frac{\pi}{5.364}$-orthogonal with probability 1.

Table 1. Near-orthogonality of RI-based Word Spaces

$\beta-$ orthogonal	Case-1 ($\varepsilon=2$)			Case-2 ($\varepsilon=4$)			Case-3 ($\varepsilon=6$)		
	$d=20$	$d=30$	$d=40$	$d=20$	$d=30$	$d=40$	$d=20$	$d=30$	$d=40$
$\pi/5.364$	-	-	-	-	-	-	1	1	1
$\pi/4.347$	-	-	-	1	1	1	-	-	-
$\pi/3.735$	-	-	-	-	-	-	0.997	1	1
$\pi/3$	1	1	1	0.984	0.995	0.998	0.958	0.992	0.997
$\pi/2.552$	-	-	-	-	-	-	0.769	0.921	0.964
$\pi/2.383$	-	-	-	0.830	0.920	0.954	-	-	-
$\pi/2.239$	-	-	-	-	-	-	0.391	0.654	0.783
$\pi/2$	0.818	0.875	0.905	0.387	0.552	0.649	0.085	0.234	0.356
Distinct Words	100	225	400	2025	11025	36100	14400	207025	1299600

Table 1 illustrates the effect of dimension of index vectors on β-orthogonality of RI-based Word Spaces for the three cases discussed above. Irrespective of the dimension of index vector, the Word Spaces of Case-1, Case-2 and Case-3 are $\frac{\pi}{3}$-orthogonal, $\frac{\pi}{4.347}$-orthogonal and $\frac{\pi}{5.364}$-orthogonal respectively with probability 1. For a fixed dimension d and any $\beta \in (0, \pi/2]$, β-orthogonality is achieved with highest probability for $\varepsilon = 2$ (Case-1); and probability of β-orthogonality decreases as ε increases. But for a smaller value of ε, one has to select a very large d to accommodate all the words of a large document. If one prefers a small d to accommodate all the words of the same document, a higher value of ε is to be chosen. Hence there is a trade off between parameters ε and d of index vectors and β-orthogonality of Word Space. The question therefore arises how this trade off actually affects the quality of summaries of RI-based extractive summarizer. Section 5 deals with this issue in detail.

5 EFFECT OF NEAR-ORTHOGONALITY ON THE PERFORMANCE OF RISUM

To study the effect of near-orthogonality on the performance of RISUM, we have used DUC 2002 [19] corpus consisting of newswire documents on various topics as experimental dataset. DUC 2002 is the last version of DUC that included single-document summarization evaluation of informative summaries. Later DUC editions contained a single-document summarization task as well; however, only very short summaries (e.g. headline summaries) were analyzed**.** Since our work is not focused on producing headline summaries, we considered DUC 2002 corpus as our experimental dataset. Each document of DUC 2002 corpus is accompanied by two different abstracts manually created by professionals. These summaries serve as the 'gold' summaries of the corresponding document. The output summary produced by RISUM for each document is limited to 10% of the length of the original document. These summaries are termed as 'system' summaries of the document and were evaluated using ROUGE (Recall Oriented Understudy for Gisting Evaluation) [8] toolbox. ROUGE measures summary quality by counting the overlapping units, such as, *n*-gram, word sequences, word pairs between the system summary and gold summaries. According to the recommendations of Lin [8] for single document summarization, we have used ROUGE-2, ROUGE-L, ROUGE-W-1.2, and ROUGE-S* for summary evaluation. We have considered 95% confidence interval for the evaluation.

5.1 CASE-1: RI WITH ONE '+1' AND ONE '-1'

In this case the dimension of index vector d_{RI} is determined using equation (20):

$$d_{RI} = 2\lceil\sqrt{m}\rceil \tag{20}$$

where m is the number of distinct words in the document and $\lceil\ \rceil$ stands for the *ceiling* function. In fact d_{RI} is the minimum dimension that is required to accommodate all the distinct words of the document. Table 2 shows the ROUGE F-measure scores for RISUM_Cosine[3] and RISUM_Euclidean[4] with dimension of index vectors d_{RI} and $2d_{RI}$. There is almost no change in scores of RISUM_Cosine when the dimension of index vector is doubled, but there is a very little improvement in the scores of RISUM_Euclidean. Also, it can be observed RISUM_Euclidean performs better than RISUM_Cosine irrespective of dimensions of index vectors.

Table 2. ROUGE F-measure scores for Case-1 (RI with one '+1' and one '-1')

Evaluation Measure	RISUM_Cosine		RISUM_Euclidean	
	d_{RI}	$2d_{RI}$	d_{RI}	$2d_{RI}$
ROUGE-2	0.132	0.129	0.155	0.159
ROUGE-L	0.222	0.221	0.265	0.268
ROUGE-W-1.2	0.145	0.145	0.169	0.171
ROUGE-S*	0.108	0.108	0.132	0.137

[3] *RISUM_Cosine: RISUM with cosine dissimilarity measure*

[4] *RISUM_Euclidean: RISUM with Euclidean distance measure*

5.2 CASE-2: RI WITH TWO '+1'S AND TWO '-1'S

In this case d_{RI}, the minimum dimension of index vector required to accommodate all the distinct words of a document is determined using equation (21):

$$d_{RI} = 2\left\lceil 0.5\left(1+\sqrt{1+8\sqrt{m}}\right)\right\rceil \quad (21)$$

Table 3 shows the ROUGE F-measure scores for RISUM_Cosine and RISUM_Euclidean with dimension of index vectors d_{RI} and $2d_{RI}$. When dimension of index vectors taken as d_{RI}, RISUM_Euclidean performs much better than RISUM_Cosine. When dimension of index vectors is increased by two times, performance of RISUM_Cosine improved significantly. However, as in Case-1 above, only a negligible improvement can be noticed in the performance of RISUM_Euclidean. Also, it can be observed with dimension $2d_{RI}$ ROUGE scores of RISUM_Cosine is still inferior to corresponding ROUGE scores of RISUM_Euclidean.

Table 3. ROUGE F-measure scores for Case-2 (RI with two '+1's and two '-1's)

Evaluation Measure	RISUM_Cosine		RISUM_Euclidean	
	d_{RI}	$2d_{RI}$	d_{RI}	$2d_{RI}$
ROUGE-2	0.102	0.127	0.157	0.161
ROUGE-L	0.195	0.221	0.268	0.275
ROUGE-W-1.2	0.126	0.143	0.170	0.175
ROUGE-S*	0.084	0.102	0.136	0.142

5.3 CASE-3: RI WITH THREE '+1'S AND THREE '-1'S

In Case-3, for each document the minimum dimension of index vector d_{RI} is twice of the real root of the equation (22):

$$x^3 - 3x^2 + 2x - 6\sqrt{m} = 0 \quad (22)$$

Table 4 shows the ROUGE F-measure scores for RISUM_Cosine and RISUM_Euclidean with dimension of index vectors d_{RI} and $2d_{RI}$. When dimension of index vectors is taken as d_{RI}, RISUM_Euclidean performs much better than RISUM_Cosine. When dimension of index vectors increased twofold, performance of RISUM_Cosine improved significantly; but surprisingly there is a slight degradation in the performance of RISUM_Euclidean. Also, it can be observed with dimension $2d_{RI}$ ROUGE scores of RISUM_Cosine is still inferior to corresponding ROUGE scores of RISUM_Euclidean.

Table 4. ROUGE F-measure scores for Case-3 (RI with three '+1's and three '-1's)

Evaluation Measure	RISUM_Cosine		RISUM_Euclidean	
	d_{RI}	$2d_{RI}$	d_{RI}	$2d_{RI}$
ROUGE-2	0.064	0.106	0.150	0.147
ROUGE-L	0.149	0.199	0.268	0.260
ROUGE-W-1.2	0.096	0.129	0.169	0.165
ROUGE-S*	0.049	0.087	0.133	0.128

5.4 COMPARISON OF ROUGE SCORES FOR THE THREE CASES

Tables 2, 3 and 4 help us in comparing the performance of the RISUM_Cosine and RISUM_Euclidean algorithms. The findings may be summarized as follows:

1. When dimension of index vector is fixed at d_{RI} for respective cases, the performance of RISUM_Cosine degrades gradually as we move from Case-1 to Case-3; but performance of RISUM_Euclidean remains almost the same for the three cases.

2. Increasing the dimension to $2d_{RI}$improves the performance of RISUM_Cosine in all the three cases. Moreover, the scores in Case-2 with respect to the four ROUGE measures become almost at par with the respective scores of Case-1; while the scores in Case-3 still remains inferior.
3. Increasing the dimension to $2d_{RI}$ does not have a very significant effect on the performance of RISUM_Euclidean unlike in the case of RISUM_Cosine. For Case-1 and Case-2 marginal improvements can be noticed in all the four ROUGE measures, but in Case-3 there is a marginal fall in the scores.

Table 5, Figures 2 and 3 provide more insight into the effect of near-orthogonality on the performance RISUM_Cosine scheme. Table 5 shows the probabilities of $\pi/3$-orthogonality and $\pi/2$-orthogonality of the respective RI-based Word Spaces against the average values of d_{RI} and $2d_{RI}$ of index vector for DUC 2002 experimental dataset. Figures 2 and 3 respectively show the effect of $\pi/3$-orthogonality and $\pi/2$-orthogonality on the performance of RISUM_Cosine.

1. As the probability of $\pi/3$-orthogonality of the RI-based Word Space increases, the performance of RISUM_Cosine also improves. The improvement in ROUGE scores can be noticed till the probability of $\pi/3$-orthogonality reaches 0.99.
2. As the probability of $\pi/2$-orthogonality of the RI-based Word Space increases from 0.003 to 0.146, steady improvement can be observed in the performance of RISUM_Cosine. The performance RISUM_Cosine falls marginally as the probability of $\pi/2$-orthogonality increases from 0.146 to 0.227. Further, increase in probability of $\pi/2$-orthogonality from 0.227 to 0.526 shows improved performance of RISUM_Cosine. Increase in probability of $\pi/2$-orthogonality beyond 0.526 does not show any effect on the performance of RISUM_Cosine.

Table 5. Near-orthogonality of RI-based Word Space over DUC 2002 experimental dataset

β-orthogonal	Case-1		Case-2		Case-3	
	$d_{RI} = 32$	$2d_{RI} = 64$	$d_{RI} = 14$	$2d_{RI} = 28$	$d_{RI} = 12$	$2d_{RI} = 24$
$\pi/3$	1	1	0.955	0.994	0.707	0.979
$\pi/2$	0.882	0.939	0.227	0.526	0.003	0.146

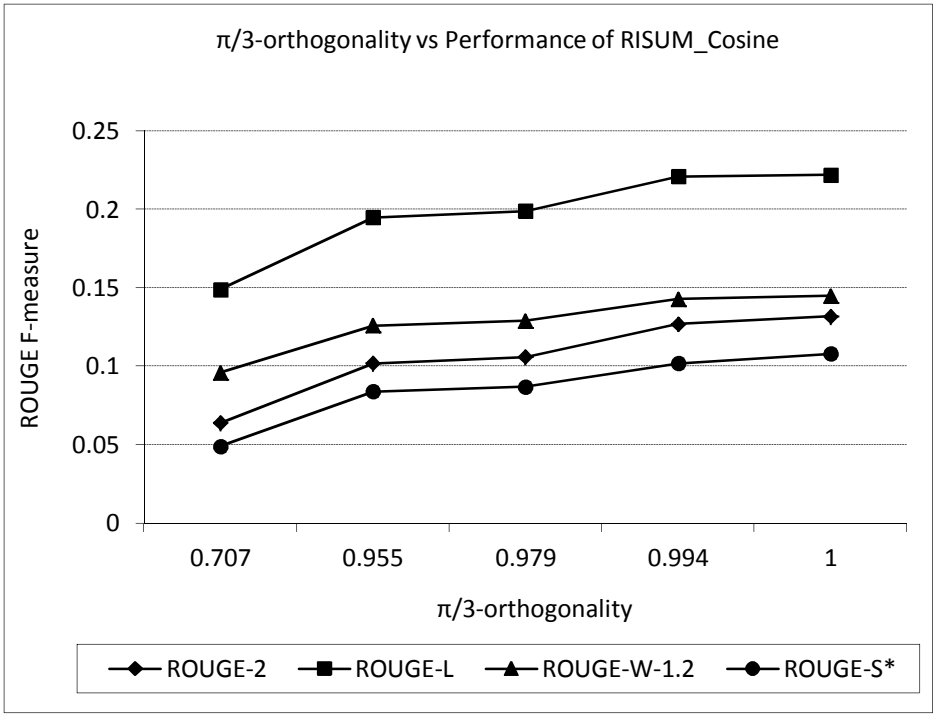

Fig. 2. Effect of π/3-orthogonality on the performance of RISUM_Cosine

From the above discussions it can be concluded that, the increase in performance of RISUM_Cosine is directly proportional to the increase in the probability of $\pi/3$-orthogonality of the RI-based Word Space. But, the probability of $\pi/2$-orthogonality has no relational effect in the performance of RISUM_Cosine. From the study it has been inferred that if one can ensure $\pi/3$-orthogonality with probability more than 0.99 in an RI-based Word Space, then very high-quality performance can be achieved for RISUM_Cosine. Further, it has been observed that the performance of RISUM_Euclidean is mostly invariant of near-orthogonality of the Word Space. This is because in RISUM_Euclidean the proximity measure is the linear distance unlike the angular distance in RISUM_Cosine. Since orthogonality and near-orthogonality are angular properties it does not affect the performance of RISUM_Euclidean.

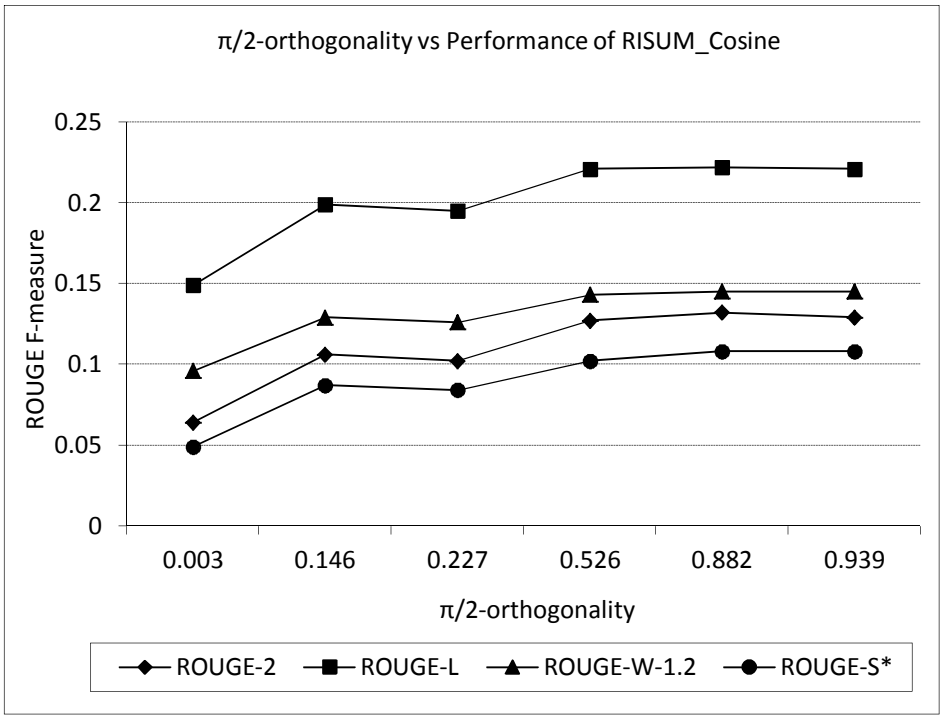

Fig. 3. Effect of $\pi/2$-orthogonality on the performance of RISUM_Cosine

5.5 COMPARISON OF RISUM WITH LSA-BASED SUMMARIZERS

In this section we compare the performance of RISUM with different LSA-based summarizers proposed in literatures. For comparison the ROUGE-L F-measure scores of LSA-based summarizers are taken from Ozsoy *et al.* [14], where 10% summaries of DUC 2002 dataset were produced. Each LSA-based summarizer was evaluated against different weighting schemes. For comparison purpose we have taken the best performance score of the each of the summarizers. Also, the best performance scores of RISUM_Cosine and RISUM_Euclidean were considered for comparison. ROUGE-L F-measure scores of the summarizers are given in Table 6. It has been observed that RISUM_Cosine performs better than cross method of Ozsoy *et al.* [10], [14], and at par with the approach of Steinberger and Ježek [12]. Also, it has been observed that RISUM_Euclidean performs much better than all LSA-based summarizers. The lowest ROUGE-L F-measure score of RISUM_Euclidean is 0.260 (Case-3, $2d_{RI}$), which is much higher than the best scores of all LSA-based summarizers.

The consistently better performance of RISUM_Euclidean over all LSA-based summarization techniques is significant because of two reasons. Firstly, the size of index vector in RI is much lower than the size of the word vector in LSA. Secondly, RISUM avoids complex dimension reduction technique, like SVD, used by LSA-based approaches – which make them computationally expensive.

Table 6. ROUGE-L F-measure scores of summarizers

Summarizer	ROUGE-L F-measure score
Gong and Liu (binary)	0.234
Steinberger and Ježek (binary)	0.224
Murray *et al.* (binary)	0.230
Ozsoy *et al.* (cross method) (binary)	0.196
Ozsoy *et al.* (topic method) (binary)	0.230
RISUM_Cosine (Case-1, d_{RI})	0.222
RISUM_Euclidean (Case-2, $2d_{RI}$)	0.275

6 CONCLUSION

This paper explores near-orthogonality – a significant aspect of the RI-based Word Space. We formulated a definition of near-orthogonality for RI-based Word Spaces. This definition should be equally applicable to other types of Word Spaces as well. We compared the near-orthogonality in RI-based Word Spaces by varying the number of '+1's and '-1's, and dimension of the index vectors. The effect of near-orthogonality on RISUM, an extractive summarization scheme based on RI, has been evaluated for these cases. Two variations of the above scheme have been studied: RISUM_Cosine and RISUM_Euclidean. Our experiments reveal that the RISUM_Cosine performs best when random index vector with one '+1' and one '-1' is considered. Construction of Word Space with index vectors having more numbers of '+1's and '-1's, requires choosing a dimension for the index vectors such that the Word Space should be $\pi/3$-orthogonal with probability more than 0.99. This will ensure a high-quality performance of the RISUM_Cosine. However, the performance of RISUM_Euclidean is almost invariant of near-orthogonality of the Word Space. Also, RISUM_Euclidean performs much better than the LSA-based summarization techniques proposed in literature.

The focus of the present work has been on extraction based single document summarization using RI. In future we plan to focus on multi-document summarization using RI, and study the effect of near-orthogonality in this context.

REFERENCES

[1] N. Chatterjee and S. Mohan, "Extraction-based Single-Document Summarization using Random Indexing," In: *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, vol. 2, pp. 448-455, 2007.

[2] M. Sahlgren, "An Introduction to Random Indexing," In: *H. Witschel (ed.): Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering (TKE 2005)*, 2005.

[3] M. Sahlgren, "*The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-dimensional Vector Spaces*," PhD Dissertation, Stockholm University, Sweden, 2006.

[4] H. Schütze, "Word Space," In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS 1993)*, pp. 895-902, 1993.

[5] T. K. Landauer and S. T. Dumais, "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge," *Psychological Review*, vol. 104, no. 2, pp. 211-240, 1997.

[6] T. K. Landauer, P. W. Foltz, and D. Laham, "Introduction to Latent Semantic Analysis," *Discourse Processes*, vol. 25, pp. 259-284, 1998.

[7] N. Chatterjee and P. K. Sahoo, "Near-orthogonality of Random Index Vectors and its Effect on Extractive Text Summarization," *Proceedings of the Indian National Science Academy (On the Occasion of the Golden Jubilee of IIT Delhi, India)*, vol. 77, no. 2, pp. 207-218, 2011.

[8] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," In: *Proceedings of the ACL-04 Workshop on Text Summarization Branches Out*, pp. 74-81, 2004.

[9] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. 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.

[10] M. G. Ozsoy, I. Cicekli, and F. N. Alpaslan, "Text Summarization of Turkish Texts using Latent Semantic Analysis," In: *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pp. 869-876, 2010.

[11] 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 (SIGIR 2001)*, pp. 19-25, 2001.

[12] J. Steinberger and K. Ježek, "Using Latent Semantic Analysis in Text Summarization and Summary Evaluation," In: *Proceedings of the 5th International Conference on Information Systems Implementation and Modelling (ISIM 2004)*, pp. 93-100, 2004.

[13] G. Murray, S. Renals, and J. Carletta, "Extractive Summarization of Meeting Recordings," In: *Proceedings of the 9th European Conference on Speech Communication and Technology*, 2005.

[14] M. G. Ozsoy, I. Cicekli, and F. N. Alpaslan, "Text Summarization using Latent Semantic Analysis," *Journal of Information Science*, vol. 37, no. 4, pp. 405-417, 2011.

[15] M. F. Porter, "An Algorithm for Suffix Stripping," *Program*, vol. 14, no. 3, pp. 130-137, 1980.

[16] D. Higgins and J. Burstein, "Sentence Similarity Measures for Essay Coherence," In: *Proceedings of the 7th International Workshop on Computational Semantics (IWCS 2007)*, 2007.

[17] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pp. 404-411, 2004.

[18] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107-117, 1998.

[19] Document Understanding Conference 2002. [Online] Available: http://www-nlpir.nist.gov/projects/duc/data.html (accessed August 2012).