Extraction-Based Single-Document Summarization Using Random Indexing

Niladri Chatterjee
Department of Mathematics
Indian Institute of Technology Delhi
New Delhi, India – 110016
niladri@maths.iitd.ac.in

Shiwali Mohan
Division of Instrumentation and Control
Netaji Subhas Institute of Technology
New Delhi, India - 110075
shiwali.mohan@gmail.com

Abstract

This paper presents a summarization technique for text documents exploiting the semantic similarity between sentences to remove the redundancy from the text. Semantic similarity scores are computed by mapping the sentences on a semantic space using Random Indexing.

Random Indexing, in comparison with other semantic space algorithms, presents a computationally efficient way of implicit dimensionality reduction. It involves inexpensive vector computations such as addition. It thus provides an efficient way to compute similarities between words, sentences and documents. Random Indexing has been used to compute the semantic similarity scores of sentences and graph-based ranking algorithms have been employed to produce an extract of the given text.

1. Introduction

Automatic Text Summarization is an important and challenging area of Natural Language Processing. The task of a text summarizer is to produce a synopsis of any document or a set of documents submitted to it.

Summaries differ in several ways. A summary can be an *extract* i.e. certain portions (sentences or phrases) of the text is lifted and reproduced verbatim, whereas producing an *abstract* involves breaking down of the text into a number of different key ideas, fusion of specific ideas to get more general ones, and then generation of new sentences dealing with these new general ideas [1]. A summary can be of a single document or multiple documents, *generic* (author's perspective) or *query oriented* (user specific) [2], *indicative* (using keywords indicating the central topics) or *informative* (content laden) [3].

In this work we have focused on producing a generic, extractive, informative, single document summary exploiting the semantic similarity of sentences.

2. Previous Work in Extractive Text Summarization

Various methods have been proposed to achieve extractive summarization. Most of them are based on scoring of the sentences. Maximal Marginal Relevance [4] scores the sentences according to their relevance to the query, Mutual Reinforcement Principle for Summary generation [5] uses clustering of sentences to score them according to how close they are to the central theme. QR decomposition method [6] scores the sentences using column pivoting. The sentences can also be scored by certain predefined features. These features may include linguistic features and statistical features, such as location, rhetorical structure [7], presence or absence of certain syntactic features [8] and presence of proper names, and statistical measures of term prominence [9].

Rough set based extractive summarization [10] has been proposed that aims at selecting important sentences from a given text using rough sets, which has been traditionally used to discover patterns hidden in data.

Methods using similarity between sentences and measures of prominence of certain semantic concepts and relationships [11] to generate an extractive summary have also been proposed.

Some commercially available extractive summarizers like Copernic [12] and Word [13] summarizers use certain statistical algorithms to create a list of important concepts and hence generate a summary.

1082-3409/07 $25.00 © 2007 IEEE
DOI 10.1109/ICTAI.2007.28

We propose to achieve extractive summarization as a three-step process:
1. Mapping the words and sentences onto a semantic space (Word Space)
2. Computing similarities between them.
3. Employing the use of graph-based ranking algorithms [14] to remove the redundant sentences in the text.

The task involves simple mathematical computations, such as addition of vectors, and thus is far more effective than other algorithms based on semantic similarities, such as LSA based summarization that involves expensive matrix computations.

3. The Word Space Model

The Word-Space Model [15] is a spatial representation of word meaning. The complete vocabulary of any text (containing n words) can be represented in an n-dimensional space in which each word occupies a specific point in the space and has a vector associated with it defining its meaning.

The Word Space Model is based entirely on language data available. It does not rely on previously compiled lexicons or databases to represent the meaning. It only represents what is really there in the current universe of discourse. When meanings change, disappear or appear in the data at hand, the model changes accordingly. If an entirely different set of language data is used, a complete different model of meaning is obtained.

3.1. The Word Space Hypotheses

The Word Space is so constructed that the following two hypotheses hold true.

1. *The Proximity Hypothesis:* The words which lie closer to each other in the word space have similar meanings while the words distant in the word space have dissimilar meanings.

2. *The Distributional Hypothesis:* Once the language data is obtained, the word space model uses the statistics about the distributional properties of the words. The words which are used within similar group of words (i.e. similar context) should be placed nearer to each other.

3.2. Context Vectors and Co-occurrence matrices

Once the distributional property of a word is obtained, the next step is to transform the distributional information into a geometric representation. In other words "*The distribution of an element will be understood as the sum of all its environments*" [16].

An environment in linguistics is called a context. A context of a word can easily be understood as the linguistic surrounding of the word. As an illustration, consider the following sentence.

'A friend in need is a friend indeed'

If we define the context of a focus word as one preceding and one succeeding word, then the context of '*need*' is '*in*' and '*is*', whereas the context of '*a*' is '*is*' and '*friend*'. To tabulate this context information a co-occurrence matrix of the following form can be created, in which the (i,j)th element denotes the number of times word i occurs in the context of word j within the text.

	Co-occurrents					
Word	a	friend	in	need	is	indeed
a	0	2	0	0	1	0
friend	2	0	1	0	0	1
in	0	1	0	1	0	0
need	0	0	1	0	1	0
is	1	0	0	1	0	0
indeed	0	1	0	0	0	0

Here the context vector for 'need' is [0 0 1 0 1 0] and for 'a' is [0 2 0 0 1 0]. They effectively sum up the environments (contexts) of the words in question, and can be represented in a six-dimensional geometric space (since the text contains 6 words). A context vector thus obtained can be used to represent the distributional information of the word into geometrical space. (Note that this is similar to assigning a unary index vector to '*is*' ([0 0 0 0 1 0]) and to '*in*' ([0 0 1 0 0 0]) and adding them up to get the context vector of '*need*'.)

3.3. Similarity in Mathematical Terms

Context vectors as such do not convey any beneficial information. They just give the location of the word in the word space. To get a clue on 'how similar the words are in their meaning' a similarity measure of the context vectors is required.

Various schemes, such as scalar product of vectors, Euclidean distance, Minkowski metrics [15], can be used to compute similarity between vectors.

We have used cosine of the angles between the two vectors *x* and *y* to compute normalized vector similarity. The cosine angle between two vectors *x* and *y* is defined as:

$$\mathrm{sim}_{\mathrm{COS}}(x,y) = \frac{x \cdot y}{|\,x\,||\,y\,|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

The cosine measure corresponds to taking the scalar product of the vectors and then dividing by their norms. The cosine measure is the most frequently utilized similarity metric in word-space research. The advantage of using cosine metric over other metrics to calculate similarity is that it provides a fixed measure of similarity, which ranges from 1 (for identical vectors), to 0 (for orthogonal vectors) and -1 (for vectors pointing in the opposite directions). Moreover, it is also comparatively efficient to compute.

3.4. Problems Associated with Implementing Word Spaces

The dimension *n* used to define the word space corresponding to a text document is equal to the number of unique words in the document. The number of dimensions increases as the size of text increases. Thus a text document containing a few thousands of words will have a word space of few thousands of dimensions. Thus computational overhead increases rapidly with the size of the text. The other problem is of data sparseness. The majority of cells in co-occurrence matrix constructed corresponding to the document will be zero. The reason is that the most of the words in any language appear in limited context, i.e. the words they co-occur with are very limited.

The solution to this predicament is to reduce the high dimensionality of the vectors. A few algorithms attempt to solve this problem by dimensionality reduction. One of the simplest ways is to remove words belonging to certain grammatical classes. Other way could be employing Latent Semantic Analysis [17]. We have used Random Indexing [18] to address the problem of high dimensionality.

4. Random Indexing

Random Indexing (RI) [18] is based on Pentti Kanerva's [19] work on sparse distributed memory. Random Indexing was developed to tackle the problem of high dimensionality in word space model. While dimensionality reduction does make the resulting lower-dimensional context vectors easier to compute with, it does not solve the problem of initially having to collect a potentially huge co-occurrence matrix. Even implementations that use powerful dimensionality reduction, such as SVD [17], need to initially collect the words-by-documents or words-by-words co-occurrence matrix. RI removes the need for the huge co-occurrence matrix. Instead of first collecting co-occurrences in a co-occurrence matrix and then extracting context vectors from it, RI incrementally accumulates context vectors, which can then, if needed, be assembled into a co-occurrence matrix.

4.1. RI Algorithm

Random Indexing accumulates context vectors in a two step process:

1. Each word in the text is assigned a unique and randomly generated vector called the *index vector*. The index vectors are sparse and high dimensional and ternary (i.e. 1, -1, 0). Each word is also assigned an initially empty *context vector* which has the same dimensionality (r) as the index vector.
2. The context vectors are then accumulated by advancing through the text one word taken at a time, and adding the context's index vector to the focus word's context vector. When the entire data has been processed, the r-dimensional context vectors are effectively the sum of the words' contexts.

For illustration we can again take the example of the sentence

'A friend in need is a friend indeed'

Let the dimension r of the index vector be 10 for illustration purposes. The context is defined as one preceding and one succeeding word.
Let *'friend'* be assigned a random index vector:

[0 0 0 1 0 0 0 0 -1 0]

and *'need'* be assigned a random index vector:

[0 1 0 0 -1 0 0 0 0 0]

Then to compute the context vector of *'in'* we need to sum up the index vector of its context. Since the context is defined as one preceding and one succeeding word, the context of *'in'* is *'friend'* and *'need'*. We sum up their index vectors to get the context vector of *'in'*.

[0 1 0 1 -1 0 0 0 -1 0]

If a co-occurrence matrix has to be constructed, r-dimensional context vectors can be collected into a matrix of order w x r, where w is the number of unique word types, and r is the chosen dimensionality of for each word.

Note that this is similar to constructing an n-dimensional unary context vector which has a single 1 in different positions for different words and n is the number of distinct words. Mathematically, these n dimensional unary vectors are orthogonal, whereas the r-dimensional random index vectors are nearly orthogonal. There are many more nearly orthogonal than truly orthogonal directions in a high-dimensional space [18]. Choosing random indexing is an advantageous tradeoff between the number of dimensions and orthogonality, as the r-dimensional random index vectors can be seen as approximations of the n-dimensional unary vectors.

Observe that both the unary vectors and the random index vectors assigned to the words construct the word space. The context vectors computed on the language data are used in mapping the words onto the word space. In our work we used Random Indexing because of the advantages discussed below.

4.2. Advantages of Random Indexing

Compared to other word space methodologies Random Indexing approach is unique in the following three ways:

First, it is an incremental method, which means that the context vectors can be used for similarity computations even after just a few examples have been encountered. By contrast, most other word space methods require the entire data to be sampled before similarity computations can be performed.

Second, it uses fixed dimensionality, which means that new data do not increase the dimensionality of the vectors. Increasing dimensionality can lead to significant scalability problems in other word space methods.

Third, it uses implicit dimension reduction, since the fixed dimensionality is much lower than the number of words in the data. This leads to a significant gain in processing time and memory consumption as compared to word space methods that employ computationally expensive dimension reduction algorithms.

4.3. Assigning Semantic Vectors to Documents

The average term vector $\vec{\mathbf{x}}_{mean}$ can be considered as the central theme of the document and is computed as:

$$\frac{1}{n}\sum_{i=1}^{n}\vec{\mathbf{x}}_i = \vec{\mathbf{x}}_{mean}$$

where *n* is the number of distinct words in the document.

While we compute the semantic vectors for the sentences we subtract $\vec{\mathbf{x}}_{mean}$ from the context vectors of the words of the sentence to remove the bias from the system [21]. The semantic vector of a sentence is thus computed as:

$$\frac{1}{n}\sum_{i=1}^{n}(\vec{\mathbf{x}}_i - \vec{\mathbf{x}}_{mean})$$

where, *n* is the number of words in the focus sentence and *i* refers to the i^{th} word of the sentence and $\vec{\mathbf{x}}_i$is the corresponding context vector.

Note that subtracting the mean vector reduces the magnitude of those term vectors which are close in direction to the mean vector, and increases the magnitude of term vectors which are most nearly opposite in direction from the mean vector. Thus the words which occur very commonly in a text, such as the auxiliary verbs and articles, will have little influence on the sentence vector so produced. Typically, these words do not have any definitive pattern about the words they co-occur with. Further, the terms whose distribution is most distinctive will be given the most weight.

5. The Experimental Setup

Our experimental data set consists of fifteen documents containing 200 to 300 words each. The processing of each document to generate a summary has been carried out as follows:

5.1. Mapping of Words onto the Word Space

Each word in the document was initially assigned a unique randomly generated index vector of the dimension 100 with ternary values (1, -1, 0). This provided an implicit dimensionality reduction of around 50%. The index vectors were so constructed that each vector of 100 units contained two randomly placed 1 and two randomly placed -1s, rest of the units were assigned 0 value. Each word was also assigned an initially empty context vector of dimension 100. The dimensions r assigned to the words depend upon the number of unique words in the text. Since our test data consisted of small paragraphs of 200-300 words each the vector of dimensions 100 sufficed. If larger texts containing thousands of word are to be summarized larger dimensional vectors have to be employed.

We defined the context of a word as two words on either side. Thus a 2x2 sliding window was used to accumulate the context vector of the focus word. The context of a given word was also restricted in one sentence, i.e. across sentence windows were not considered. In case where the window extended in the preceding or the succeeding sentence, a unidirectional window was used. There is fair evidence supporting the use of small context window. Kaplan [22] conducted various experiments with people in which they successfully guessed the meaning of a word if two words on either side of it were also provided. Experiments conducted at SICS, Sweden [23] also indicate that a narrow context window is preferable for acquiring semantic information. The above observation prompted us to use a 2x2 window. The window can be weighted as well to give greater importance to the words lying closer to the focus word. For example, the weight vector [0.5 1 0 1 0.5] suggests that the words adjacent to the focus word are given the weight 1 and the words at distance 2 are assigned a weight of 0.5. In our experiments we have used the above mentioned weights for computing the context vectors.

5.2. Mapping of Sentences onto the Word Space

Once all the context vectors have been accumulated, semantic vectors for the sentences were computed. A mean vector was calculated from the context vectors of all the words in the text. This vector was subtracted from the context vectors of the word appearing in the sentence, the resultants were summed up and averaged to compute the semantic vector of the sentence.

5.3. Construction of Completely Connected Undirected Graph

We constructed a weighted, completely connected, undirected graph from the text, wherein each sentence is represented by a node in the graph. The edge joining node *i* and node *j* is associated with a weight w_{ij} signifying the similarity between the sentence *i* and sentence *j*.

5.3.1. Assigning the Node Weights

The weight of each node was determined by calculating the 'relevance' of each sentence in the text. For this purpose, we identified the index words [9] of the document. An index word is a word which has a frequency higher than a predetermined lower cutoff and does not belong to the grammatical classes like articles, prepositions and auxiliary verbs. All the index words were assigned a weight which was calculated by dividing the number of occurrence of the word by total number of distinct words in the text.

For example in the text containing 6 distinct words: *"A friend in need is a friend indeed"* the word *'friend'* occurs twice. Thus it is assigned a weight of 2/6 = 0.333.

A 1-dimensional vector was allocated to each sentence, with length equal to the number of index words and each element referring to an index word. The value of that element was determined by multiplying the number of times the index word occurred in the sentence by its weight.

An average document vector was calculated by averaging the context vectors of all its sentences. Cosine similarity of each of the sentence with the document vector was calculated. The value thus obtained was assigned to the respective node and will be called the node weight.

5.3.2. Assigning the Edge Weights

The edges between the nodes are weighted by the similarity scores between the participating nodes (sentences). The similarity was computed by determining the cosine metric between the sentence vectors.

5.4 Calculating Weights of the Sentences and Generating Summary

Once the graph is constructed, our aim is to get rid of the redundant information in the text by removing the sentences of less importance. To achieve this, the sentences are ranked by applying some graph-based ranking algorithms. Various graph-based ranking algorithms are available in literature. The one that we have used for this work is weighted PageRank algorithm [24].

Weighted PageRank Algorithm

Let G = (V, E) be a directed graph with the set of vertices V and set of edges E, where E is a subset of VxV. For a given vertex V_i, let In(V_i) be the set of vertices that point to it (predecessors), and let Out(V_i) be the set of vertices that vertex V_i points to (successors). Then the new node weight assigned by PageRank ranking algorithm after one iteration is:

$$PR^W(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} w_{ji} \frac{PR^W(V_j)}{\sum\limits_{V_k \in Out(V_j)} w_{kj}}$$

This computation is carried out on all the nodes in succession iteratively until node weights converge. We set the value 0.85 to the factor d as per the recommendation in [25]. We have applied this algorithm on undirected graphs constructed for each text by considering In(V_i) = Out(V_i) for all nodes.

The node weights converge in such a way that weights of the important sentences are highly increased, while those of the redundant sentences do not increase in same proportion. Once the weights stabilize, the sentences are sorted in descending order of their weights. The top few sentences are selected to generate the summaries.

6. Results

We have run the experiments on different texts and computed extracts at 10%, 25%, 50% levels. We compared the results with manual summaries created by experts and also the summaries generated by some commercially available summarizers namely Copernic and Word. Below we show a sample text and its summaries at different percentage levels generated by our scheme. We also show the summaries generated by Copernic and Word at different levels and also the expert generated summary.

Consider the text given below as our sample document, which is a ten-sentence long document. As mentioned earlier, for evaluation purpose we have used documents which are 200 to 300 words long.

A solar eclipse occurs when the Moon passes between Earth and the Sun, thereby totally or partially obscuring Earth's view of the Sun.

This configuration can only occur during a new moon, when the Sun and Moon are in conjunction as seen from the Earth.

In ancient times, and in some cultures today, solar eclipses are attributed to mythical properties.

Total solar eclipses can be frightening events for people unaware of their astronomical nature, as the Sun suddenly disappears in the middle of the day and the sky darkens in a matter of minutes.

However, the spiritual attribution of solar eclipses is now largely disregarded.

Total solar eclipses are very rare events for any given place on Earth because totality is only seen where the Moon's umbra touches the Earth's surface.

A total solar eclipse is a spectacular natural phenomenon and many people consider travel to remote locations in order to observe one.

The 1999 total eclipse in Europe, said by some to be the most-watched eclipse in human history, helped to increase public awareness of the phenomenon.

This was illustrated by the number of people willing to make the trip to witness the 2005 annular eclipse and the 2006 total eclipse.

The next solar eclipse takes place on September 11, 2007, while the next total solar eclipse will occur on August 1, 2008.

The sentences selected by experts manually to create a summary are:

A solar eclipse occurs when the Moon passes between Earth and the Sun, thereby totally or partially obscuring Earth's view of the Sun.

This configuration can only occur during a new moon, when the Sun and Moon are in conjunction as seen from the Earth.

Total solar eclipses are very rare events for any given place on Earth because totality is only seen where the Moon's umbra touches the Earth's surface.

A total solar eclipse is a spectacular natural phenomenon and many people consider travel to remote locations in order to observe one.

The 1999 total eclipse in Europe, said by some to be the most-watched eclipse in human history, helped to increase public awareness of the phenomenon.

The summary generated by our summarizer:

10% summary

A solar eclipse occurs when the Moon passes between Earth and the Sun, thereby totally or partially obscuring Earth's view of the Sun.

25% summary

A solar eclipse occurs when the Moon passes between Earth and the Sun, thereby totally or partially obscuring Earth's view of the Sun.

This configuration can only occur during a new moon, when the Sun and Moon are in conjunction as seen from the Earth.

Total solar eclipses can be frightening events for people unaware of their astronomical nature, as the Sun suddenly disappears in the middle of the day and the sky darkens in a matter of minutes.

50% summary

A solar eclipse occurs when the Moon passes between Earth and the Sun, thereby totally or partially obscuring Earth's view of the Sun.

This configuration can only occur during a new moon, when the Sun and Moon are in conjunction as seen from the Earth.

Total solar eclipses can be frightening events for people unaware of their astronomical nature, as the Sun suddenly disappears in the middle of the day and the sky darkens in a matter of minutes.

Total solar eclipses are very rare events for any given place on Earth because totality is only seen where the Moon's umbra touches the Earth's surface.

The 1999 total eclipse in Europe, said by some to be the most-watched eclipse in human history, helped to increase public awareness of the phenomenon.

The summary generated by Copernic:

10% summary

A solar eclipse occurs when the Moon passes between Earth and the Sun, thereby totally or partially obscuring Earth's view of the Sun.

25% summary

A solar eclipse occurs when the Moon passes between Earth and the Sun, thereby totally or partially obscuring Earth's view of the Sun.

A total solar eclipse is a spectacular natural phenomenon and many people consider travel to remote locations in order to observe one.

50% summary

A solar eclipse occurs when the Moon passes between Earth and the Sun, thereby totally or partially obscuring Earth's view of the Sun.

total solar eclipses can be frightening events for people unaware of their astronomical nature, as the Sun suddenly disappears in the middle of the day and the sky darkens in a matter of minutes.

A total solar eclipse is a spectacular natural phenomenon and many people consider travel to remote locations in order to observe one.

The 1999 total eclipse in Europe, said by some to be the most-watched eclipse in human history, helped to increase public awareness of the phenomenon.

The summary generated by Word:

10% summary

The next solar eclipse takes place on September 11, 2007, while the next total solar eclipse will occur on August 1, 2008.

25% summary

A solar eclipse occurs when the Moon passes between Earth and the Sun, thereby totally or partially obscuring Earth's view of the Sun.

A total solar eclipse is a spectacular natural phenomenon and many people consider travel to remote locations in order to observe one.

The next solar eclipse takes place on September 11, 2007, while the next total solar eclipse will occur on August 1, 2008.

50% summary

A solar eclipse occurs when the Moon passes between Earth and the Sun, thereby totally or partially obscuring Earth's view of the Sun.

A total solar eclipse is a spectacular natural phenomenon and many people consider travel to remote locations in order to observe one.

The 1999 total eclipse in Europe, said by some to be the most-watched eclipse in human history, helped to increase public awareness of the phenomenon.

This was illustrated by the number of people willing to make the trip to witness the 2005 annular eclipse and the 2006 total eclipse.

The next solar eclipse takes place on September 11, 2007, while the next total solar eclipse will occur on August 1, 2008.

For larger texts, we used Precision, Recall and *F*, widely used in Information Retrieval [26] for evaluating our results. For each document an extract done manually by experts has been considered as the reference summary (denoted by *Sref*). We then compare the candidate summary (denoted by *Scand*) with the reference summary and compute the precision, recall and *F* values as follows:

$$p = \frac{|S_{ref} \bigcap S_{cand}|}{S_{cand}} \quad r = \frac{|S_{ref} \bigcap S_{cand}|}{S_{ref}} \quad F \ = \frac{2pr}{p+r}$$

We also compute the precision, recall and F values for the summaries generated by Copernic [12] and Word summarizer [14] by comparing them with *Sref*. Finally we compare the p, r, F values corresponding to our summarizer with these values. The values obtained for fifteen documents have been tabulated in Table 1.

Text Number		**Our Approach**			**Copernic**			**Word**		
		p	r	F	p	r	F	p	r	F
1	10%	1.000	0.166	0.284	1.000	0.083	0.154	1.000	.250	0.400
	25%	0.800	0.333	0.424	0.500	0.166	0.252	0.800	.333	0.424
2	10%	1.000	0.444	0.444	1.000	0.140	0.250	0.500	0.142	0.222
	25%	1.000	0.429	0.601	1.000	0.429	0.545	0.333	0.142	0.200
3	10%	0.500	.125	0.200	0.500	0.125	0.200	0.500	0.125	0.200
	25%	0.600	.375	0.462	0.750	0.375	0.500	0.600	0.375	0.462
4	10%	1.000	0.200	0.333	1.000	0.200	0.333	0	0	N.A.
	25%	1.000	0.400	0.570	0.666	0.400	0.498	0.666	0.400	0.498
5	10%	1.000	0.143	0.249	1.000	0.143	0.249	0.500	0.143	0.222
	25%	0.750	0.426	0.545	0.666	0.286	0.400	0.500	0.286	0.329
6	10%	1.00	0.300	0.451	1.000	0.200	0.333	0.500	0.100	0.167
	25%	0.833	0.500	0.625	0.666	0.300	0.400	0.400	0.200	0.267
7	10%	1.000	0.222	0.364	1.000	0.222	0.364	0.500	0.111	0.181
	25%	0.200	0.444	0.552	0.750	0.333	0.458	0.400	0.400	0.282
8	10%	1.000	0.250	0.400	1.000	0.250	0.400	0	0	N.A.
	25%	1.000	0.500	0.666	1.000	0.500	0.666	0.500	0.250	0.400
9	10%	0.750	0.200	0.315	0.666	0.133	0.221	0.500	0.133	0.210
	25%	0.875	0.466	0.608	0.857	0.400	0.545	0.625	0.333	0.434
10	10%	0.666	0.200	0.307	1.000	0.200	0.333	0.500	0.200	0.285
	25%	0.833	0.500	0.625	0.800	0.400	0.533	0.714	0.500	0.588
11	10%	1.000	0.166	0.285	1.000	0.166	0.284	0.666	0.166	0.265
	25%	0.857	0.500	0.632	0.666	0.333	0.444	0.875	0.583	0.699
12	10%	0.666	0.125	0.211	1.000	0.125	0.222	0.666	0.125	0.210
	25%	0.875	0.438	0.583	0.750	0.375	0.500	0.750	0.375	0.500
13	10%	1.000	0.222	0.363	1.000	0.222	0.363	0.666	0.222	0.333
	25%	0.800	0.444	0.571	0.750	0.333	0.461	0.600	0.333	0.428
14	10%	1.000	0.182	0.305	1.000	0.182	0.308	0.600	0.182	0.279
	25%	0.833	0.454	0.593	0.800	0.364	0.499	0.600	0.364	0.453
15	10%	1.000	0.230	0.373	1.000	0.230	0.373	0.666	0.154	0.249
	25%	0.857	0.461	0.599	0.666	0.307	0.419	0.714	0.384	0.499

Table 1: Results of Our summariser, Copernic and Word Summarizer

The observations clearly indicate that the summaries generated by our method are closer to the human generated summaries that the summaries produced by Copernic and Word summarizers at 10% and 25% level in almost all the text cases. At 50% level too we obtained better results compared to Copernic and Word. However, limitation of space precludes us to show the figures in this table.

6. Conclusions and Future Scope

In this paper we have proposed a summarization technique which involves mapping of the words and sentences onto a semantic space and exploiting their similarities to remove the less important sentences containing redundant information. The problem of high dimensionality of the semantic space corresponding to the text has been tackled by employing Random Indexing which is less expensive in computations and memory consumption compared to other dimensionality reduction approaches. The approach gives better results than commercially available summarizers namely Copernic and Word Summarizer.

In future we plan to include a training algorithm using Random Indexing which will construct the Word Space on a previously compiled text database and then to use it for summarization purposes so as to resolve the ambiguities, such as polysemy, more efficiently.

We observed some abruptness in the summaries generated by our method. We plan to smoothen out this abruptness by constructing Stiener trees of the graphs constructed corresponding to the text.

In our present evaluation we have used measures like precision, recall and F which are used primarily in the context of information retrieval. In future we intend to use more summarization-specific techniques, e.g. ROUGE [27] to measure the efficacy of our scheme.

Text summarization is an important challenge in present day context for huge volumes of text are being produced every day. We expect that the proposed approach will paves the way for developing an efficient AI tool for text summarization.

7. References

[1] Inderjeet Mani, "Advances in Automatic Text Summarization", *MIT Press*, Cambridge, MA, USA, 1999.

[2] Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell, "Summarizing text documents:sentence selection and evaluation metrics", *ACM SIGIR*, 1999, pp 121–128.

[3] E.H. Hovy and C.Y. Lin, "Automated Text Summarization in SUMMARIST", *Proceedings of the Workshop on Intelligent Text Summarization, ACL/EACL-97*. Madrid, Spain, 1997.

[4] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," *ACM SIGIR*, 1998, pp. 335–336.

[5] Zha Hongyuan, "Generic Summarization and Key phrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering", *ACM*, 2002.

.

[6] John Conroy, Leary Dianne, "Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition", *ACM SGIR* ,2001.

[7] Daniel Marcu "From discourse structures to text summaries" In *ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, 1997, pp 82–88.

.

[8] J. Pollock and A. Zamora "Automatic abstracting research at chemical abstracts service", *JCICS*, 1975.

[9] Hans P. Luhn, "The automatic creation of literature abstracts", *IBM J. of R. and D*, 1958.

[10] Nidhika Yadav, M.Tech Thesis , Indian Institute of Technology Delhi, 2007

[11] Yihong Gong and Xin Liu, "Generic text summarization using relevance measure and latent semantic analysis", *SIGIR,ACM,* 2001, pp 19–25.

[12] Copernic Summarizer Homepage: http://www.copernic .com /en/products/summarizer.

[13] Rada Mihalcea and Paul Tarau, "An Algorithm for Language Independent Single and Multiple Document Summarization", *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, Korea, October 2005.

[14] Word Sumamriser www.microsoft.com/education/ autosummarize.mspx

[15] M. Sahlgren, "The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces", Ph.D. dissertation, Department of Linguistics, Stockholm University, 2006

[16] Z. Harris, *Mathematical structures of language*, Interscience Publishers, 1968.

[17] Thomas K Landauer, Peter W. Foltz, Darrell Laham, "An Introduction to Latent Semantic Analysis", *45th Annual Computer Personnel Research Conference – ACM*, 2004.

[18] M. Sahlgren, "An Introduction to Random Indexing. Proceedings of the Methods and Applications of Semantic Indexing", *Workshop at the 7th International Conference on Terminology and Knowledge Engineering*, TKE, Copenhagen, Denmark, 2005.

[19] P. Kanerva, *Sparse distributed memory*, Cambridge, MA, USA: MIT Press, 1988

[20] S. Kaski, "Dimensionality reduction by random mapping: Fast similarity computation for clustering", Proceedings *of the International Joint Conference on Neural Networks, IJCNN'98* IEEE Service Center, 1999.

[21] Derrick Higgins, Jill Burstein, "Sentence similarity measures for essay coherence", *Proceedings of the 2004 Human language Technology Conference of the North American chapter of the Association for Computational Linguistics*, Boston, Massachusetts, May 2004

[22] A. Kaplan "An experimental study of ambiguity and context", *Mechanical Translation*, 2(2), 1955.

[23] J. Karlgren, & M. Sahlgren (2001), "From words to understanding", *Foundations of real-world intelligence* ,CSLI Publications, 2001

[24] Rada Mihalcea, "Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization", *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, companion volume (ACL 2004)*, Barcelona, Spain, July 2004.

[25] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine", *Computer Networks and ISDN Systems*, 1998.

[26] R.B. Yates, B.R. Neto, *Modern Information Retrieval,* Pearson Education, 1999

[27] ROUGE: Recall Oriented Understudy for Gisting evaluation, http://www.isi.edu/~cyl/ROUGE/