Multi-document Text Summarization Using Random Indexing and Clustering

Ambrish Rawat[a], Pramod K. Sahoo[b], Niladri Chatterjee[a]

[a] *Department of Mathematics, Indian Institute of Technology Delhi, New Delhi, India*
[b] *Institute for Systems Studies & Analyses, Defence Research & Development Organisation, New Delhi, India*

Abstract

Extractive text summarization for a set of documents suffers from two serious problems, viz. repetition of information in several documents, complementary information existing on the same theme across different documents. As a consequence, algorithms applicable for single document extractive summarization do not work well in multi-document scenario. In this paper we have proposed a clustering based technique where the sentences of the document collection are clustered based on their themes. The sentences in each cluster are then ranked according to their importance. The ranked sentences are then picked in a systematic manner to generate the summary. Random Indexing, which has assumed significant attention in recent times, has been used as the representation scheme for the documents in order to avoid large binary vectors as is the case with typical Word Space Model. The system has been tested with DUC 2007 dataset. The proposed algorithm is found to perform better than some existing algorithms as per the scores given by various ROUGE metrics.

Keywords: Multi-document Text Summarization, Random Indexing, Clustering, Sentence Extraction

1. Introduction

In extractive text summarisation representative sentences are chosen from a document for generation of the summary [1]. One of the ways of achieving this is to represent sentences in a multi-dimensional space. Word Space Model

(WSM) is a technique where vector representation of sentences is used for representing a given text in a Word Space [2]. Typically, these vectors are of very large dimension. Hence an efficient representation of text using WSM is computationally intensive for a large set of documents. One of the most common WSM-based approach is Latent Semantic Analysis (LSA) ([3],[4]), where orthogonal unary vectors are used to represent the words of a document. The dimension of these vectors is equal to the number of unique words in the document. The sentence vectors are then computed as weighted sums of the word vectors. However, large dimension of the vectors make them computationally challenging. Although Singular Value Decomposition (SVD) is often used for dimensionality reduction, it renders the whole procedure computationally more expensive.

Random Indexing [5] provides a scalable alternative to LSA where near-orthogonal *index vectors* are used for achieving an incremental learning of the context. Moreover, by fixing a lower dimension for the index vectors used for word representation, one avoids the computationally expensive task of SVD. Random Indexing has already been used for extractive text summarisation of single documents in [6] and [7]. An improved formulation of Random Indexing based single document extractive summarisation has also been suggested in [8].

A straightforward application of single document summarization strategies in a multi-document scenario, however, suffers from some serious problems. Some of them are:

- The size of the document collection may be very large in comparison to a single document. For instance, one particular collection in the DUC2007 data set comprised of 24 documents with roughly 22000 words making the summary generation a challenging task.

- A set of documents may have a common theme. For instance, in the collection, D0701, of DUC2007 data set, many documents have a sentence which in some form conveys the information that "Morris Dees is the co-founder of the Southern Poverty Law Center" (Figure 1). A summa-

rizer for this collection should not capture these different sentences as it will result in redundant information in the generated summary. A multi-document summarizer should therefore have the ability to minimize this redundant information present across its document collection.

Document number	Text on Morris Dees
APW20000909.0144	Morris Dees, co-founder of the Southern Poverty Law Center in Montgomery, Ala., represented the Keenans and has said he intends to take everything the Aryan Nations owns to pay the judgment, including the sect's name.
APW20000908.0184	Morris Dees, the co-founder of the Southern Poverty Law Center in Montgomery, Ala., and one of the attorneys for the plaintiffs, said he intended to enforce the judgment, taking everything the Aryan Nations owns, including its trademark name.
NYT19980715.0137	In the same issue, Morris Dees, leader of the Southern Poverty Law Center, gets an unwarranted slap in the MediaWatch column. Dees has always gotten props for dropping dimes on the hateful activities of such groups as the Ku Klux Klan and Aryan Nation.

Figure 1: Excerpts from three documents of D0701 collection from DUC 2007 Dataset

- It is also important that a multi-document text summarizer fuses different information about a common prevalent theme across different documents. For example, many documents in D0701 describe some piece of information about "the Southern Poverty Law Center" in different contexts (Figure 2). A summarizer for this collection should be able to capture this as a theme and include different sentences pertaining to this theme but focussing on different aspects of the theme.

A significant difference between single-document and multi-document extractive text summarisation stems from the redundancy of chosen sentences and the compression ratio of the summary. Since the degree of redundancy among the sentences from a set of related documents is much higher than among the sentences from a single document, it is essential that multi-document text summarisers adopt measures to address redun-

Document number	Text on Southern Poverty Law Center
NYT19990302.0069	But the book didn't begin, nor will it end with the King trial, as a report by the Montgomery, Ala.-based Southern Poverty Law Center demonstrates. According to the report, the number of hate groups in the country increased again last year.
NYT19990304.0376	Since co-founding the Southern Poverty Law Center in 1971, Dees has wielded the civil lawsuit like a buck knife, carving financial assets out of hate group leaders who inspire followers to beat, burn and kill.
NYT19990822.0227	Lawyers from the Southern Poverty Law Center, a civil rights organization in Montgomery, Ala., advanced his case in federal court, charging that the Escude Funeral Home and Hixson Brothers Funeral Home in Avoyelles Parish either refused to deal with blacks or offered ``distinctly inferior services'' for the same prices that they charged whites.

Figure 2: Excerpts from three documents of D0701 collection from DUC 2007 Dataset

dancy.

Two generic approaches in this regard can be found in literature:

- By measuring the similarity of new candidate sentences with respect to the previously selected ones, and pick the one that is least similar to the previously selected ones [9].
- Another approach is by clustering the sentences from the document collection and thereby capturing the multiple themes present in the collection [10]. A summary can thereafter be generated by selecting representative sentences from different clusters.

The present paper proposes a hybrid approach of the Random Indexing based semantic representation along with traditional clustering techniques to achieve a summary for a given set of documents. In section 2, we describe the general principles of Random Indexing. Section 3 discusses the proposed strategy which is a combination of the Random Indexing based representation

proposed for single-document summariser and the clustering algorithms. The experimental results have been summarised in section 4 and the conclusion and future work have been briefed in section 5.

2. Random Indexing

Random Indexing aims to assign near-orthogonal *index vectors* of fixed dimension to a set of words for a smaller and denser representation in the Word Space. These index vectors are randomly assigned in a manner such that they are unique (i.e each distinct word gets a different index vector), sparse and ternary. Every vector consists of values – 0, +1 and -1, which are randomly assigned according to the following distribution,

$$\begin{cases} +1 & \text{with probability } \frac{n_\epsilon}{2d_I} \quad (1a) \\ 0 & \text{with probability } \frac{d_I - n_\epsilon}{d_I} \quad (1b) \\ -1 & \text{with probability } \frac{n_\epsilon}{2d_I} \quad (1c) \end{cases}$$

Here, d_I is the fixed dimension of the set of index vector and $n_\epsilon \ll d_I$. We have conducted experiments with different values of n_ϵ for random-indexing based single document text summarisation, where the value of d_I depends on the size of the document [11]. In the present work, we have experimented with three different values of n_ϵ viz. $1, 2$ and 3 and correspondingly three different values of d_I as suggested in [11]. With an assumption that the context of a word is determined by its neighbouring words, the context vectors for all the words are calculated by adding the index vectors of neighbouring words. In our experimental framework, we have defined the neighbourhood of a word as consisting of the two words succeeding and the two words preceding the word under consideration.

3. Clustering based multi-document text summarisation

The proposed strategy for text summarisation consists of the following three steps:

- Representation of sentences of a document collection as *sentence semantic vectors* in the Word Space using the index vectors as described in Section 2.

- Clustering of the sentence semantic vectors in the proximity graph. After mapping the sentences into the Word Space, the entire document is converted into a proximity graph. In this graph the sentences are represented as nodes, and edge weights between two sentences give the strength of semantic proximity between sentences.

- Selection of representative sentence(s) from the ranked clusters and summary generation.

In this work, we have experimented with a combination of different index vector assignments along with a combination of two different clustering strategies (explained in Section 3.2), and two different sentence selection strategies (explained in Section 3.3) for subsequent summary generation. We also experimented with three different distance measures in the clustering and sentence selection strategies.

3.1. Representation of sentences in the Word Space

Semantic vector representation of the sentences in a document collection has been achieved in four steps -

- preprocessing of the available textual data from the collection,

- generation of index vectors for the unique words,

- computation of context vectors for each unique word in the collection, and finally

- generation of sentence semantic vectors for each sentence in the document collection.

Preprocessing: As a first step, the texts from all the documents in a document collection are combined into one single text file. This is followed by a

combination of text-processing steps: removal of common words and stemming. A set of commonly occurring words is extracted from the combined document to get a list of *content words* The Porter Stemming Algorithm [12] is then used to remove common morphological inflectional endings from the obtained *content words*. Further computations are carried out on this updated list of *unique content words* $\{u_i\}$.

Generation of Index Vectors: This is done as per the scheme explained in section 2. A unique ternary index vector, $I(u_i)$, is generated for each word, u_i, in the *unique content words* list for a given value of n_ϵ. Here, n_ϵ, is defined as the number of '+1' (or '-1') assigned to every index vector. In this work we have experimented and compared results for three different values of n_ϵ viz. 1, 2 and 3. As per the suggestions in [11], the dimension of the index vector, d_I has been varied according to both n_ϵ and m, where m is the total number of *unique content words*.

$$d_I = \begin{cases} 2\lceil\sqrt{m}\rceil, & n_\epsilon = 1 \quad (2a) \\ 2\lceil 0.5(1+\sqrt{1+8\sqrt{m}})\rceil, & n_\epsilon = 2 \quad (2b) \\ \text{real root of } x^3 - 3x^2 + 2x - 6\sqrt{m}, & n_\epsilon = 3 \quad (2c) \end{cases}$$

Here, $\lceil\ \rceil$ is the ceiling function. In each of the above equations (3a, 3b, 3c) d_I denotes the minimum dimension required for generating RI based index vectors for all the distinct words of a document.

One difficulty in purely random assignment of +1 and -1 is that during the context vector calculation, through the summation of index vectors, some +1 may get cancelled with some -1 leaving an erroneous context for the word. To avoid this, we have restricted the placement of the '+1'(s) to the upper half of an index vector and the placement of the '-1'(s) has been restricted to the lower half (as suggested in [13]).

Generation of Context Vectors: A context vector for each word in the *unique content words* list is iteratively generated be adding index vectors of its neigh-

bourhood. As mentioned in Section 2, we have considered the two succeeding and the two preceding words along with the word under consideration, for the neighbourhood definition. This gives us the following formulation for the context vector, $C(w_{i,j})$, for the word, $w_{i,j}$ occurring at i^{th} position of the j^{th} sentence of the combined document,

$$C(w_{ij}) = C(w_{ij}) + \sum_{k=-2}^{2} 2^{1-|k|} I(w_{(i-k)j}) \tag{3}$$

where $I(w_{ij})$ is the index vector of w_{ij} and $2^{1-|k|}$ is the weighting factor of the neighbouring word which accounts for its proximity to the word under consideration.

Generation of Sentence Semantic Vectors: The sentence semantic vector, denoted as $S(s_j)$, for j^{th} sentence is defined as,

$$S(s_j) = \frac{1}{m_j} \sum_{i=1}^{m_j} (C(w_{ij}) - T) \tag{4}$$

where m_j is the number of content words in j^{th} sentence and T denotes the central theme of the document. The central theme of the document collection is computed by taking the arithmetic mean of context vectors of all the content words in the given document collection.

$$T = \frac{1}{m} \sum_{i=1}^{m} (C(u_i)) \tag{5}$$

Subtraction of central theme from the documents takes care of the commonly occurring words which unlike in context vector computation are not ignored here [14]. The sentence semantic vectors thus computed provide a geometrical representation of the semantic being conveyed by the sentence. A Semantic vector in itself does not convey much information. It merely represents the location of sentences in the word space. It is the relative distance between the sentences that captures the similarity in the meaning of sentence. Two sentences, (as represented by their semantic vectors) that lie close to each other in the Word Space have similar meaning.

3.2. Clustering the sentences in the proximity graph

With each sentence assigned a semantic vector in the Word Space, the document collection is represented by the proximity graph. The nodes of the graph represent the sentences of the entire document collection. The distance between two nodes is representative of the similarity in theme and meanings of sentences being represented by those nodes. We exploit this relative thematic similarity of closely spaced sentences in the clustering strategies. These strategies capture the multiple themes present in the document collection as a set of clusters, $\mathbf{N} = \{N_1 ... N_{n_c}\}$, where n_c equals the total number of clusters. In this work In this work we have used DUC 2007 as our gold standard data; and experimented with four different values of n_c namely,

$$n_c = 5, 10, 15, 20 \tag{6}$$

We employed two traditional unsupervised clustering techniques – K-Means Clustering [15] and Clustering based on Expectation Maximization of Gaussian Mixture Models (EMGM-Clustering [16]). Each of these strategies utilises different mathematical framework while clustering the data points. We have used three different distance measures namely, Euclidean distance, cosine dissimilarity and angular distance - for calculating the distance between any pair vectors, v_i and v_j, in the Word Space. Equations (7), (8) and (9) give the mathematical formulation of the three distance metrics.

$$D(v_i, v_j) = \sqrt{\sum_{t=1}^{d} (v_{it} - v_{jt})^2} \tag{7}$$

$$D(v_i, v_j) = 1 - \frac{\sum_{t=1}^{d} (v_{it} v_{jt})}{\sqrt{\sum_{t=1}^{d} (v_{it})^2} \sqrt{\sum_{t=1}^{d} (v_{jt})^2}} \tag{8}$$

$$D(v_i, v_j) = \frac{\arccos \frac{\sum_{t=1}^{d} (v_{it} v_{jt})}{\sqrt{\sum_{t=1}^{d} (v_{it})^2} \sqrt{\sum_{t=1}^{d} (v_{jt})^2}}}{\pi} \tag{9}$$

3.2.1. K-Means Clustering

In this strategy the association of a sentence semantic vector $S(s_i)$ with a cluster is iteratively computed such that the inter-cluster distance between a pair of sentence semantic vectors belonging to different clusters is maximised; while the intra-cluster distance between a pair of sentence semantic vectors belonging to the same cluster is minimised [15]. In every iteration the sentence semantic vector $S(s_i)$ is assigned to a cluster N_k such that,

$$k = \min_{\mathbf{p=1...n_c}} D(m_p, s_i) \tag{10}$$

where $\mathbf{N}$ is the set of all clusters and m_p is the *cluster centroid* of p^{th} cluster N_p which is defined as the arithmetic mean of all sentence semantic vector belonging to that cluster. With the varying vector associations, the *cluster centroid* , m_p, for a cluster N_p is updated in each iteration as,

$$m_p = \frac{1}{l} \sum_{i=1}^{l} (S(s_i)) \tag{11}$$

where l is equal to the total number of sentences belonging to the cluster N_p in the current iteration.

Thus, on convergence, a pair of sentences, s_i and s_j, with similar themes will have their sentence semantic vectors $S(s_i)$ and $S(s_j)$ in the same cluster N_r. Similarly, a pair of sentences, s_a and s_b, with similar themes will have their sentence semantic vectors $S(s_a)$ and $S(s_b)$ in different cluster N_p and N_q.

3.2.2. EMGM Clustering

In EMGM clustering we use Expectation Maximization algorithm for fitting the Gaussian Mixture Models (GMM) to cluster the sentence semantic vectors. It is assumed that the set of sentence semantic vectors have a joint distribution given by $p(S(s_i))$ with n_c Gaussian components. Thus, it is assumed that a cluster, N_k, has a multivariate gaussian distribution, $p_k(\mu_k, \Sigma_k)$, in the words space with μ_k and Σ_k as the mean vector and the covariance matrix, respectively,

of the underlying gaussian distribution.

$$p(S(s_i)|\Theta) = \sum_{k=1}^{n_c} \alpha_k p_k(S(s_i)|\mu_k, \Sigma_k) \tag{12}$$

$$p_k(S(s_i)|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d_R/2}|\Sigma_k|^{1/2}} e^{\frac{-1}{2}(S(s_i)-\mu_k)^T(\Sigma_k^{-1}(S(s_i)-\mu_k)))} \tag{13}$$

Here, Θ is the set of parameters α_k, μ_k and Σ_k. Since Θ is unknown, we estimate it by initialising it randomly, and then updating it iteratively. The update is done so as to maximise the likelihood of assignment of semantic vectors to a Gaussian sample.

Each cluster is representative of a theme from the document collection – this is denoted as the *cluster theme* T_k. The *cluster theme* for a cluster is computed as the arithmetic mean of the semantic vectors of all the sentences in that cluster.

$$T_k = \frac{1}{b_k} \sum_{i=1}^{b_k} (S(s_{ik})) \tag{14}$$

Here, b_k denotes the total number of sentence semantic vectors in k^{th} cluster, and s_{ik} is the sentence semantic vector of the i^{th} sentence in the k^{th} cluster.

Extending the reasoning from the previous section, the relative distance between cluster themes exemplifies the similarity in themes of the clusters. This gives a natural ranking to the clusters according to their distance from the central theme of the document collection. Each cluster N_p is thus given a rank $Rank(p) = k$, with $k \in \{1, 2, ..., n_c\}$, such that,

$$D(T_1, T) < D(T_2, T) < ... < D(T_k, T) < D(T_{k+1}, T) < ... < D(T_{n_c}, T) \tag{15}$$

3.3. Representative sentence selection from the ranked clusters and summary generation

In order to obtain candidate sentences for a summary, sentences are selected in a defined order from the formed clusters. It was noted that selecting multiple sentences from one cluster will lead to redundant sentences in a summary which convey similar meaning pertaining to that particular cluster's *cluster theme*. Also, selection of sentences from a cluster that is not close to the central theme in the Word Space, will lead to less relevant sentences in a summary. In order to balance the relevance and redundancy of sentences in a summary, we devised the following scheme.

Starting from cluster p with $Rank(p) = 1$, we chose the 1^{st} sentence which is the *best (sentence rank = 1) representative* of the cluster theme of that document. Similarly, the 2^{nd} sentence in the summary is the *best (sentence rank = 1) representative* of the cluster q, where $Rank(q) = 2$. The process is continued till *best (sentence rank = 1) representative* from all n_c clusters has been selected. This is followed by a selection of the 2^{nd} *best (sentence rank 2) representative* from all n_c cluster in the order of their rank and so on. In short, if the i^{th} sentence of the summary is given by $U(i)$ which was obtained as the k^{th} *best ((sentence rank = k)) representative* of cluster p (N_p^k), then the $\{i+1\}^{th}$ sentence of the summary, $U(i+1)$ is given by,

$$\begin{cases} N_r^{k+1} \mid R(r) = 1 & \text{if } R(p) = n_c \quad (16a) \\ N_q^k \mid R(q) = R(p) + 1 & \text{otherwise} \quad (16b) \end{cases}$$

The n^{th} *best representative* sentence a cluster p, (N_p^n) has been computed using the following two strategies

1. n^{th} *nearest to centroid*: We arrange the sentences in a cluster p according to their Euclidean distance from the *cluster theme*, T_p. The candidacy of a sentence in a cluster is thus determined by its proximity to its cluster's *theme*. This is in accordance with the reasoning that relative distance between a pair of semantic vectors captures the similarity in meanings of

the two vectors which implies that the closer a sentence is to its cluster's *theme*, the better is its relevance in conveying the cluster's *theme*. This results in the following relationship between the i^{th} and j^{th} *best representative* sentences of a cluster.

$$D(N_p^i, T_p) < D(N_p^j, T_p) \implies i < j \tag{17}$$

2. n^{th} *longest sentence*: In this strategy, we arrange the sentences according to their length. A length of a sentence, $L(s_i)$ is defined as the number of words (including stop words) in that sentence (s_i). A sentence with a large number of words can be assumed to have large number of content words, and will therefore have significant information with respect to the theme of that cluster. With this reasoning, the sentence with the maximum length in a cluster is considered as the *best (rank 1) representative*. This means,

$$L(N_p^i)) < L(N_p^j)) \implies i < j \tag{18}$$

Having selected the representative sentences in the manner explained above, the summary of the document collection is generated from the candidate set $\{U(1), U(2), ...\}$. A summary is obtained as the sentences $\{U(1), U(2), ...U(n)\}$, such that, $L(U(1)) + L(U(2)) + ... + L(U(n)) < 250$. A summary length of 250 was required for the ROUGE score evaluation, as the Gold standard (human generated) summaries in the corpus had a field length of 250 words.

4. Experiment

We evaluated the strategies on DUC 2007 corpus which consists of 45 document collection with 25 documents in each. Each document was of varying lengths ranging from 300 to 1700 words. Attached to each document collection are four manually created summaries. Algorithmically generated 32 summaries with 250 words each, were also available. We evaluated the following ROUGE [17] scores for our proposed strategies.,

- ROUGE-1, unigram recall measure
- ROUGE-L, longest common subsequence recall measure
- ROUGE-SU, skip bigram with unigram recall measure

As explained in Section 3, five parameters were varied during the experiment - number of '+1'(or '−1') in the upper half (lower half) of the index vector ($n_\epsilon \in \{1, 2, 3\}$), number of clusters ($n_c \in \{5, 10, 15, 20\}$), choice of clustering strategy, choice of sentence selection scheme and choice of distance measure. This resulted in 48 different summaries for each document collection. The patterns observed from these are summarised in the following sections.

4.1. Different values of n_ϵ

Three different values of n_ϵ resulted in three different dimensions for the index vectors. This led to three different sets of sentence semantic vectors for each document collection. Table 1 gives the ROUGE scores for the summaries obtained with these three sets when 15 clusters were formed using the EMGM clustering strategy and representative sentence from each cluster was selected using nearest to centroid scheme. Euclidean distance measure was used in this part of the experiment.

Table 1: ROUGE Recall scores of summaries generated for different numbers of ones in the index vector ($n_c = 5$, EMGM Clustering Strategy, Closest to centroid sentence selection scheme, Euclidean distance measure).

ROUGE metric	$n_\epsilon = 1$	$n_\epsilon = 2$	$n_\epsilon = 3$
ROUGE-1	**0.36866**	0.34896	0.34834
ROUGE-L	**0.33543**	0.32165	0.3215
ROUGE-SU	0.12266	**0.13387**	0.11298

4.2. Different number of clusters

The number of clusters were varied with an intention to optimally capture the number of themes present in the document. In most cases the difference in

number of clusters did not significantly reflect on the ROUGE scores. However, $n_c = 15$, was the optimal for most sets of summaries. Table 2 summarizes the results for his experiment.

Table 2: ROUGE Recall scores of summaries generated for different numbers of clusters ($n_\epsilon = 2$, EMGM Clustering Strategy, Closest to centroid sentence selection scheme, Euclidean distance measure).

ROUGE metric	$n_c = 5$	$n_c = 10$	$n_c = 15$	$n_c = 20$
ROUGE-1	0.34896	0.36781	**0.37428**	0.36833
ROUGE-L	0.32165	0.33745	**0.34133**	0.33644
ROUGE-SU	0.11387	0.12288	**0.12468**	0.12123

4.3. Different sentence selecting schemes

In all the combinations of different clustering strategies, different index vector generation and different number of clusters, the closest to centroid scheme outperformed the longest-sentence scheme. The results of one particular combination, where the distinction was evident has been tabulated in Table 3. While using closest to centroid strategy, euclidean distance was used as the distance measure.

Table 3: ROUGE Recall scores of summaries generated for using different sentence selection schemes ($n_\epsilon = 1$, $n_c = 15$, K-Means Clustering Strategy).

ROUGE metric	closest to centroid	longest sentence
ROUGE-1	**0.38312**	0.35443
ROUGE-L	**0.34826**	0.31867
ROUGE-SU	**0.13064**	0.11676

4.4. Different clustering strategies

As mentioned in Section 3 we experimented with two different clustering strategies. Each was these strategies tried to capture the themes in the document collection in different mathematical frameworks. For most combinations

K-Means clustering strategy generated summaries with better ROUGE scores. Table 4 summarise the results for one combination of parameters. In this part of the experiment, Euclidean distance was used as the distance measured.

Table 4: ROUGE Recall scores of summaries generated for using different clustering strategies ($n_\epsilon = 1$, $n_c = 15$, Closest to centroid sentence selection scheme).

ROUGE metric	K-Means clustering	EMGM-clustering
ROUGE-1	**0.38312**	0.36975
ROUGE-L	**0.34826**	0.33678
ROUGE-SU	**0.13064**	0.12464

4.5. Different distance measures

We experimented with three different distance measures - Euclidean distance, cosine dissimilarity and angular distance. These three measures tried to capture different types of similarity relationships amongst the sentence semantic vectors clustering strategies. The results of this experiment for summaries generated using K-Means clustering strategy and closest to centroid sentence selection strategy have been summarized in Table 5.

Table 5: ROUGE Recall scores of summaries generated for using different distance measures ($n_\epsilon = 1$, $n_c = 15$, K-Means Clustering Strategy, Closest to centroid sentence selection scheme).

ROUGE metric	Euclidean	Cosine dissimilarity	Angular
ROUGE-1	**0.38312**	0.37144	0.36757
ROUGE-L	**0.34826**	0.34494	0.34273
ROUGE-SU	**0.13064**	0.12325	0.12342

5. Conclusion

This paper proposed and examined a combination of strategies for multi-document text summarisation. The summarisation strategies were based on different clustering and sentence picking schemes. Additionally, the paper also

examined the effects of varying the parameters of number of clusters. The effect of index vector with different numbers of '+1' and '−1' on the summary generation was also analysed. It was observed that the K-Means clustering strategy generated summaries with better ROUGE scores for all combinations of other parameters. Similarly it was observed that closest to centroid sentence selection scheme outperformed the longest-sentence scheme. This can be explained for K-Means clustering as in this case the clustering, cluster ranking and sentence selection strategies are distance based and utilise the same distance measure (Euclidean distance) in their computations.

Ouyang et al [18] proposed an interesting approach for multi-document summarization using different regression models. Although query-focused in its approach, the work provides for a much needed performance benchmark for multi-document summarization. Regression models have been implemented using Support Vector regressors. A set of predefined features is used to characterize different aspects of a sentence. Similarity between a sentence and a query is measured in terms of these feature values. For each sentence s of the document collection the feature values are computed, and a score is assigned indicating its importance. Regression models are then trained on to obtain the coefficients for the best scoring function, which in turn helps to choose sentences from unknown document set according to their importance. The above approach has been used on DUC 2007 datasets. It has been found that the regression based approach performed better than different participating teams. The average ROUGE-2 scores as obtained by the system have been recorded as 0.1133, and average ROUGE-SU4 recorded as 0.1652. The results of the proposed approach are comparable if not better in some cases as is clear from the ROUGE scores.

The promising performance of the use of clustering-based strategies for text multi-document text summarization encourages us to further explore the schemes with the use of different ranking schemes for sentence selection. Furthermore, we also wish to explore other sentence selection schemes which emphasise on redundancy of the chosen sentences along with their relevancy.

References

[1] I. Mani, Automatic summarization, John Benjamins Publishing, 2001.

[2] H. Schütze, Word space, in: Advances in Neural Information Processing Systems 5, Citeseer, 1993.

[3] T. K. Landauer, S. T. Dumais, A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge., Psychological review 104 (2) (1997) 211.

[4] T. K. Landauer, P. W. Foltz, D. Laham, An introduction to latent semantic analysis, Discourse processes 25 (2-3) (1998) 259–284.

[5] M. Sahlgren, An introduction to random indexing, in: Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE, Vol. 5, 2005.

[6] N. Chatterjee, S. Mohan, Extraction-based single-document summarization using random indexing, in: Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on, Vol. 2, IEEE, 2007, pp. 448–455.

[7] P. Gustavsson, A. Jönsson, Text summarization using random indexing and pagerank, in: Proceedings of the third Swedish Language Technology Conference (SLTC-2010), Linköping, Sweden, 2010.

[8] N. Chatterjee, P. K. Sahoo, Effect of near-orthogonality on random indexing based extractive text summarization, International Journal of Innovation and Applied Studies 3 (3) (2013) 701–713.

[9] J. Carbonell, J. Goldstein, The use of mmr, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1998, pp. 335–336.

[10] D. R. Radev, H. Jing, M. Styś, D. Tam, Centroid-based summarization of multiple documents, Information Processing & Management 40 (6) (2004) 919–938.

[11] P. K. Sahoo, Extractive text summarization using random indexing, Ph.D. thesis, Indian Institute of Technology Delhi (October 2013).

[12] M. F. Porter, An algorithm for suffix stripping, Program: electronic library and information systems 14 (3) (1980) 130–137.

[13] N. Chatterjee, P. K. Sahoo, Random indexing and modified random indexing based approach for extractive text summarization, Computer Speech & Language 29 (1) (2015) 32–44.

[14] D. Higgins, J. Burstein, Sentence similarity measures for essay coherence, in: Proceedings of the 7th International Workshop on Computational Semantics, 2007, pp. 1–12.

[15] J. A. Hartigan, M. A. Wong, Algorithm as 136: A k-means clustering algorithm, Applied statistics (1979) 100–108.

[16] R. A. Redner, H. F. Walker, Mixture densities, maximum likelihood and the em algorithm, SIAM review 26 (2) (1984) 195–239.

[17] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, 2004, pp. 74–81.

[18] Y. Ouyang, W. Li, S. Li, Q. Lu, Applying regression models to query-focused multi-document summarization, Information Processing & Management 47 (2) (2011) 227–237.