Sentence Clustering-based Summarization of Multiple Text Documents

Kamal Sarkar
Computer Science & Engineering Department, Jadavpur University, Kolkata – 700 032, INDIA,
[jukamal2001@yahoo.com]

Abstract:- **With the rapid growth of the World Wide Web, information overload is becoming a problem for an increasingly large number of people. Automatic Multi-document summarization can be an indispensable solution to reduce the information overload problem on the web. This kind of summarization facility helps users to see at a glance what a collection is about and provides a new way of managing a vast hoard of information. The clustering-based approach to multi-document text summarization can be useful on the web due to its domain independence and language independence nature. The clustering based multi-document summarization performance heavily depends on three important factors: (1) clustering sentences, (2) cluster ordering, (3) selection of representative sentences from the clusters. The objective of this study is to find out the suitable algorithms for sentence clustering, cluster ordering and representative sentence selection to have a successful sentence clustering based multi-document summarization system. In this paper, we present a sentence clustering based multi-document summarization system along with a comparative study on the different variations of this system. The system performances are compared to the top performing systems participated in DUC 2004.**

Keywords: Information overload; Similarity histogram-based sentence clustering; Multi-document summarization; Domain independence; Language independence

1. INTRODUCTION

The web users are overwhelmed with a large volume of information even on a single topic returned by the traditional search engines and it is very difficult for the users to go through all the hits and find the relevant information from the collection. The human understands a single or a cluster of text documents by consuming the main themes of the documents by applying some cognitive process. Multi-document summarization is a process, which produces a condensed representation of the contents of multiple related text documents collected from heterogeneous sources for human consumption. Thus, Multi-document summarization helps human to digest the main contents of multiple related text documents very rapidly. With this kind of summarization facility, during information search on the web, users can discard a set of documents after going only through the summary (gist) of them if they are not relevant to them. Thus the total web search cost is reduced. Automatic multi-document summarization has already been used on the web to help users to understand the document clusters. For example, a number of web news services, such as Google News, NewsBlaster [1] collect and group news articles into news topics, and then produce topic wise short summary. Using these web services, the users can rapidly understand the topic they are interested in by going through the gist.

Depending on the nature of text representation in the summary, summary can be categorized as an abstract and an extract. An extract is a summary consisting of a number of salient text units selected from the input. An abstract is a summary, which represents the subject matter of the article with the text units, which are generated by reformulating the salient units selected from the input. An abstract may contain some text units, which are not present in to the input text.

Although sentence extraction method is not the usual way that humans follow while creating summaries for documents, some sentences in the documents represent some aspects of their contents to some extent. Moreover, speed will be an important factor while incorporating the summarization facility on the web. So, extraction based summarization is still useful on the web. The extractive multi-document summarization can be concisely formulated as extracting important textual units from multiple related documents, removing redundancies and reordering the units to produce the fluent summary. Redundancy is one of the important factors in multi-document summarization task. A number of systems [2][3] rank sentences based on sentence-level and word-level features. Then, it selects the top ranked sentence first and measures the similarity of the next candidate textual unit (sentence or paragraph) to the previously selected

ones and retains it only if it contains enough new (dissimilar) information. A well known such measure is maximal marginal relevance (MMR)[4].

An alternative approach to ensure good coverage and avoid redundancy is the clustering based approach that groups the similar textual units (paragraphs, sentences) into multiple clusters to identify themes of common information and selects text units one by one from clusters in to the final summary [5][6][7] [8] [9]. Each cluster consists of a group of similar text units representing a sub-topic (theme). Domain independency and language independency are the key features of the clustering based approaches to multi-document text summarization. In this paper, we present a multi-document text summarization system, which clusters sentences using a similarity histogram based sentence-clustering algorithm to identify multiple sub-topics (themes) from the input set of related documents and selects the representative sentences from the appropriate clusters to form the summary. We also investigate on the different variations of this system for comparison purposes. Our system has three major components:

- Similarity histogram based incremental sentence clustering method that groups similar sentences by keeping each cluster at a high degree of coherency. Coherency of a cluster is dynamically monitored using a concept called cluster similarity histogram [10]
- Cluster ordering scheme that orders the clusters in decreasing order based on the relevance or information richness of the clusters.
- Representative sentence selection scheme that selects one sentences from each cluster. Sentences in a cluster are given scores based on local importance (how central to the cluster) and global importance (how much of multiple sub-topics are covered by a sentence).

In summary, the key Contributions of the work in this paper are:

- Adaptation of a suitable sentence-clustering algorithm, which automatically determines the number of clusters and which is unsupervised in nature.
- Introducing a new cluster ordering algorithm, which plays a vital role when the number of clusters are not known in advance
- Introducing a new method for representative selection from the clusters.
- The proposed summarization approach can easily be ported to new domain and new language due to several reasons: (A) this system does not incorporate domain dependent features such as positional information, cue phrases, sentence length etc., (B) no stemmer is used for stemming the input.

The rest of this paper is organized as follows: in section 2, we describe related work. In section 3, we describe all components of the proposed multi-document summarization system. Section 4 and section 5 describe the evaluation method and results.

2. RELATED WORK

Many previous works on extractive summarization ranks sentences based on simple features such as their position in the text, frequency of the words they contain, or some key phrases indicating the importance of the sentences [11][12][13] and select top n sentences based on the compression ratio. In another approach to multi-document summarization, information extraction is used to identify similarities and differences across the documents in the set [14].

In the centroid based multi-document summarization [2][3], the sentences are ranked based on its similarity to the cluster centroid and a number of top-ranked sentences are selected based on the compression ratio. The centroid is defined as a pseudo-document consisting of words with TF*IDF scores greater than a predefined threshold.

Redundancy is one of the important factors in multi-document summarization. Some systems [2][3] rank sentences based on sentence-level and word-level features, selects the top most sentence first, measure the similarity of a next candidate textual unit (sentence or paragraph) to that of previously selected ones and retain it only if it contains enough new (dissimilar) information. A well known such measure is maximal marginal relevance [4].

An alternative approach to ensure good coverage and avoid redundancy is the clustering based approach that groups the similar textual units (paragraphs, sentences) into multiple clusters to identify themes of common information and selects text units one by one from clusters in to the final summary [5][6][7][8][9]. Centroid based summarization method [2][3] can be thought to be a single cluster based approach since it groups the sentences closest to the centroid in to a single cluster. But, in this paper, the clustering approach means an approach that

groups sentences in to multiple clusters. Since the centroid based summarization approach ranks sentences based on its similarity to a common centroid, the similar sentences may come close in their ranks and the redundant sentences may be selected in the summary. To remove redundancies, the re-ranking algorithm MMR [4] is used. In contrary to the centroid-based approach, the multi-cluster summarization approach divides the input set of text documents in to a number of clusters (sub-topics or themes). If clusters are sufficiently distant from each other, selection of one representative from each cluster reduces the chances of appearing redundant sentences in to the summary. Stein and Bagga [15] groups documents into clusters by clustering single document summaries, and then selects representative passages from each cluster to form a summary. Boros [7] proposes a cluster based approach to summarization which clusters sentences using the clustering method assuming that number of clusters to be formed will be specified by the user in advance. They report the poor system performance. SimFinder is a sentence-clustering algorithm [16] [17] that takes the sentences from the original document set and clusters them into several clusters, also called themes. The implementation of this sentence clustering is based on rich linguistic features and trained using a statistical decision tree, which handles set features [18]. Hardy et al. [8] selects representative passages from clusters of passages, and the system worked well in DUC2001 with some properly tuned parameters. Moens, Uyttendaele, and Dumortier [9] demonstrate that clustering algorithms based on the selection of representative objects have a definite potential for automatic summarization as well as for the recognition of the thematic structure of text. They use cosine similarity between the sentences to cluster a text into different subtopics. A predefined cosine threshold is used to cluster paragraphs around seed paragraphs (called medoids). Seeds are determined by maximizing the total similarity between the seed and the other paragraphs in a cluster. The seed paragraphs are then considered as the representatives of the corresponding subtopics, and included in the final summary.

There are few successful clustering based summarization systems in the literature. The clustering based approach discussed in [7] report poor performance. In comparison to the previous works on the clustering based multi-document summarization, we focus on how clustering algorithm and representative object selection from clusters affects the multi-document summarization performance. We have also shown that the performance of clustering based summarization can be improved by improving sentence clustering and representative sentence selection methods.

While sentences are extracted from multiple source documents, picking sentences out of context may result in incoherent summary. Ensuring coherence is difficult, because this requires some understanding of the content of each passage and knowledge about the structure of discourse. Practically, most systems follow time order and text order (passages from the oldest text appear first, sorted in the order in which they appear in the input texts) [19].

Compared to creating an extract, generation of abstract is relatively harder since the latter requires: (1) semantic representation of text units (sentences or paragraphs) in the text, (2) reformulation of two or more text units and (3) rendering the new representation in natural language. Abstractive approaches have used template based information extraction, information fusion and compression.

In fact, Abstractive summarization is not matured till date. The existing abstractive summarizers often depend on an extractive preprocessing component. The cut, paste and compression operations are performed on the output of the sentence extraction component to produce the abstract of the text [20][21][22]. The system SUMMONS [23] extracts and combines information from multiple sources and passes this information to a language generation component to produce the final summary. Some approaches use information fusion techniques to identify repetitive phrases from the clusters and the phrases are fused together to form the fluent summary [24].

The CLASSY [25] summarization system, which performed the best on task 2 in DUC 2004, consists of two major components – a Hidden Markov Model for selecting sentences from each document and a pivoted QR algorithm for generating a multi-document summary. The HMM has two kinds of states, which correspond to summary and non-summary sentences in a single document. In addition, the best number of the HMM states needs to be determined based on empirical testing, and the HMM model needs to be learned using training data. In addition to these two components, CLASSY also incorporates a linguistic component as a preprocessing stage to provide the summarization engine simplified sentences as input. So, this system is dependent on some domain and language specific sub tasks. Wan [26]

proposes an approach to multi-document summarization based on affinity graphs. Their method identifies semantic relationships between sentences and uses a graph based ranking algorithm to compute the amount of information that a subset of sentences contains. The best subset of sentences was then selected as the output summary using a greedy algorithm. Their system outperformed the best result in DUC-2004. But since this system considers the stemmed input for building affinity graph and the stemming rules varies from one language to another, we cannot say that this system is language independent. Since our approach uses neither the stemmed input nor the learning algorithm, it is less domain-dependent and language dependent than the above two approaches.

Some researches have been initiated to address the possibility of improving document summarization performance through revisions of the extractive summaries by local and global sentence compression techniques [27][28][29][30].

More deeper approaches [31] [32] exploits cohesion feature or the discourse structure by using synonyms of the words or anaphora resolution

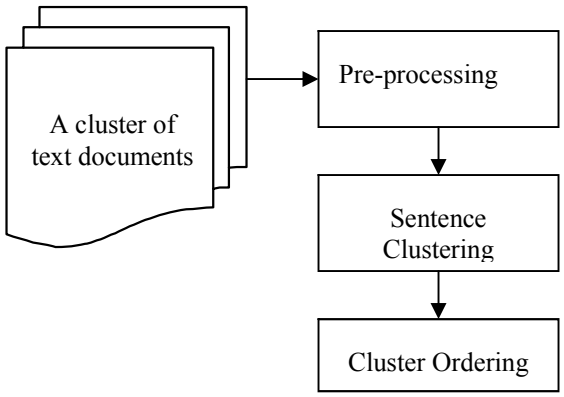
Researchers have also tried to incorporate machine learning into summarization [33][34][35][36].

3. THE PROPOSED SYSTEM ARCHITECTURE

In this section, we describe in detail the various components of the framework of the proposed system. The major components are:

- Preprocessing
- Sentence clustering
- Cluster ordering
- Representative sentence selection
- Summary generation

Fig-1 shows the framework of the proposed clustering based summarization system.



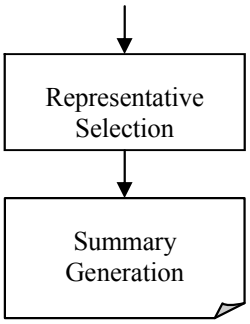


Fig 1. The framework of the proposed sentence clustering based summarization system

3.1 Preprocessing

The preprocessing task primarily includes removal of stop words (prepositions, articles and other low content words), punctuation marks (except dots at the sentence boundary).

3.2. Sentence clustering

Sentence clustering is the important component of the clustering based summarization system because sub-topics or multiple themes in the input document set should properly be identified to find the similarities and dissimilarities across the documents.

If sentences are grouped in to a predefined number of clusters, the clusters may not be coherent because some sentences may be forcibly assigned to some clusters although it should not be. The incoherent clusters may contain duplicate text units, which may lead to the selection of the redundant sentences in to summary. On the other hand, if the clusters are very tight, most of the clusters may be converted to singletons. Thus, we should have a clustering method, which ensures the coherency of the clusters and minimizes inter-cluster distance. For the sentence clustering, we adopt the similarity histogram based incremental clustering method presented in [10]. The clustering algorithm presented in [10] has been used for web document clustering. But, we adopt this method to sentence clustering. We observe that sentence-clustering task is not totally similar to the document-clustering task because the sentences are short and less informative compared to documents. We describe the details of the histogram based sentence-clustering algorithm in the section 3.2.2. One of the important factors of any clustering technique is how to compute similarity between two objects. The similarity measure

used in our sentence- clustering algorithm is discussed in the next subsection.

3.2.1 Similarity measure

Cosine similarity is a popular sentence-to-sentence similarity metric used in many clustering and summarization tasks [37][38]. Sentences are represented by a vector of weights while computing cosine similarity. But, the feature vector corresponding to a sentence becomes too sparse because sentences are too short in size compared to the input collection of sentences. Sometimes it may happen that two sentences sharing only one higher frequent word show high cosine similarity value. So, we prefer to use a uni-gram matching-based similarity measure.

$$\text{Sim}(S_i, S_j) = (2 * | S_i \cap S_j |) / (| S_i | + | S_j |) . \quad (1)$$

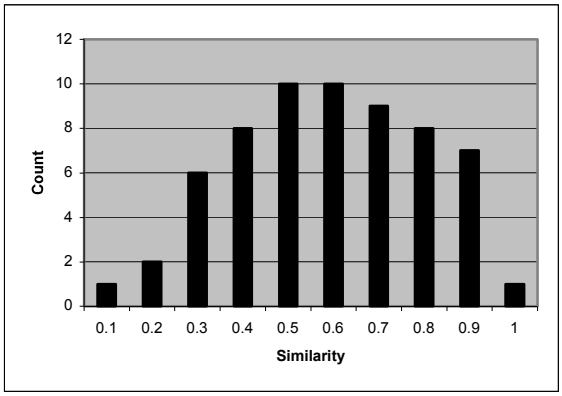
Where S_i and S_j are any two sentences belonging to the input collection of sentences. The numerator $| S_i \cap S_j |$ represents number of matching words between two sentences and
$| S_i |$ is the length of the i-th sentence, where length of a sentence =number of words in the sentence.

3.2.2 A similarity histogram based sentence clustering

This clustering approach is an incremental dynamic method of building the sentence clusters. In incremental clustering approaches, data objects are processed one at a time and data objects are incrementally assigned to their respective clusters while they progress. Incremental clustering is an essential strategy for online applications, where time is a critical factor. Our idea here is to employ an incremental sentence clustering method that will exploit our similarity measure to produce clusters of high quality.

The main concept for the similarity histogram-based clustering method is to keep each cluster as coherent as possible and a degree of coherency in a cluster at any time is monitored with a Cluster Similarity Histogram. Cluster similarity histogram is a concise representation of the set of pair-wise sentence similarities distribution in a cluster. A histogram consists of a number of bins that correspond to fixed similarity value intervals. Each bin height represents the count of pair-wise sentence similarities in the corresponding interval. In the figure 2, a typical cluster similarity histogram has been shown.

A perfect cluster would have a histogram, where all pair-wise similarities are of maximum value and the



histogram would have the right-most bin representing all similarities. On the other hand, a loose cluster would have histogram where all pair-wise similarities are minimum

Fig. 2 A cluster similarity histogram showing the distribution of pair-wise sentence similarities in a cluster

and the similarities would tend to be counted in the lower bins.

To prevent selection of the redundant sentences in to the summary, we should be careful to keep each cluster as coherent as possible. In other words, the objective would be to maximize the number of similarities in the high similarity intervals. To achieve this goal in an incremental fashion, we should judge the effect of adding a new sentence to a certain cluster. If the inclusion of this sentence is going to degrade the distribution of the similarities in the clusters very much, it should not be added, otherwise it is added.

But assignment of sentences to clusters based on similarity distribution enhancement may create problem with the perfect clusters. The sentence may be rejected by the perfect cluster even if it has high similarity to most of the sentences in the cluster. So, the quality of a similarity histogram representing cluster cohesiveness is judged by calculating the ratio of the count of similarities above a certain similarity threshold to the total count of similarities. The higher this ratio, the more coherent the cluster is.

If n be the number of the documents in a cluster, the number of pair-wise similarities in the cluster is n(n+1)/2. Let $S=\{sim_i,: i=1,..., m\}$ be the set of pair-wise sentence similarities in a cluster, where m=n(n+1)/2. The histogram of the similarities in the cluster is given by
$H=\{h_i, i=1, ..., n_b\}$
$h_i = \text{count}(sim_k) \quad sim_{li} \le sim_k \le sim_{ui,}$
where

n_b: the number of bins in a histogram
h_i: the count of sentence similarities in bin i,
sim_{li}: the lower similarity bound of bin i,
sim_{ui}: the upper similarity bound of bin i.

The histogram ratio of a cluster is calculated using the following formula:

$$\text{Histogram Ratio (HR)} = \frac{\sum_{i=T}^{n_b} h_i}{\sum_{j=1}^{n_b} h_j}, \quad \text{where}$$

$$T = \lfloor S_T * n_b \rfloor, \qquad (2)$$

S_T: the similarity threshold
T: bin number corresponding to the similarity threshold

Inclusion of a bad element in a cluster at any stage may severely affect the cluster quality and this may degrade eventually the histogram ratio to zero.

To prevent this problem, we set a minimum threshold of histogram ratio HR_L, which should at least be maintained by each cluster. The steps of the histogram based incremental sentence-clustering algorithm are shown in figure 3.

This sentence-clustering algorithm is simple to implement. It clusters sentences with no prior knowledge of the number of clusters to be formed and works only on the input sentence collection without using any corpus knowledge or linguistic knowledge. These properties of the algorithm help to keep our summarization system domain independent and language independent.

Begin
1) Say, Clist is a cluster list which is initially empty

2) Convert all the input documents in a collection of sentences, Slist. Each sentence S in Slist is indexed by the document number and the sentence number

3) For each sentence S in Slist do
For each cluster c in Clist do
3.1 Store the histogram ratio of the cluster c to a variable before adding s to c, that is, $HR_{old}=HR_c$
3.2 Simulate adding s to c to check whether addition of s to c would severely degrade or improve the histogram ratio (coherence) of c. Let the simulated histogram ratio be HR_s
3.3 If ($HR_s \geq HR_{old}$) or (($HR_s \geq HR_L$)and ($HR_{old} - HR_s <$eps))
then add s to c and exit from the inner loop to avoid any chance of assigning the same sentence to more than one cluster.
3.4 If s is not added to any cluster, then create a new cluster c,
add s to c and add c to Clist.
end-for (inner loop)
end-for(outer loop)
end

Fig. 3 Similarity histogram based incremental sentence clustering algorithm.

3.3 Cluster ordering

Since our sentence-clustering algorithm is fully unsupervised and it does not assume any prior knowledge about the number of clusters to be formed, it is crucial to decide which cluster would contribute the representative first to the summary. One simple method is to order the clusters based on their sizes measured in terms of sentence-counts assuming that the cluster which contains more number of sentences is more important. But we observe that this method does not perform well when:

- Several top clusters are of equal size
- Clusters consist of a number of less informative short sentences, which increase only the size, but not the contents.

So, to overcome this problem we have proposed a new cluster-ordering algorithm, which orders clusters based on the cluster importance, which is computed by the sum of the weights of the content words of a cluster. Instead of considering the count of sentences in a cluster as the cluster importance, we measure the importance of a cluster based on the number of important words it

contains. The importance of a cluster is computed as follows:

Weight of a cluster C, W(C) =

$$\sum_{w \in C} \log(1 + count(w)), \quad (3)$$

Where count (*w*) is the count of the word w in the input collection and the count (w) is greater than a threshold. The cluster weight represents the information richness of the cluster. Here, we weigh the clusters by the number of important (relevant) words found in the clusters to ensure that the size should truly represent the information richness of the cluster. We measure importance of a term with respect to the total input collection documents. The log-normalized value of the total count of a word in the set of input documents has been taken as the weight of a word. Before computing the counts of the words in the input collection all stop words are removed because the stop words occurs more frequently in the natural language texts and they always show higher counts.

After ordering the clusters in decreasing order of their importance, we select the top n clusters. One representative sentence is selected from each cluster and includes in to the summary. We continue selecting sentences until a predefined summary size is reached.

The next important question is how to choose a representative sentence from a cluster, which is basically a group of nearly similar sentences. But, sentences in a cluster may not be perfectly similar to each other since the similarity threshold should be set to a value less than the ideal value of 1 for achieving better clustering performance. We observe that setting similarity threshold to the ideal value of 1 degrades clustering performance because it converts most of the clusters to singletons. The issues related to representative sentence selection are discussed in the next sub-section 3.4.

3.4 Representative sentence selection

In this section, we discuss the possible answers of the question: how to select representative sentences from the clusters? We try to find the possible solutions in many ways:

- arbitrary (random) selection
- longest candidate selection
- sentence selection based on its similarity to the centroid of input document set
- Sentence selection based on local and global importance.

Ideally, the random selection can be thought to be a solution expecting that all the sentences in the cluster are perfectly similar to each other and any member of the cluster is sufficient to represent the cluster theme. But, practically, it does not happen so, because we traditionally use a similarity threshold to judge whether two sentences are similar or not. Hence, two similar sentences in clusters may share some dissimilar information. We observe that setting the similarity threshold to the highest possible value (i.e., 1) does not improve the summarization performance because it converts most of the clusters to singletons.

The second solution i.e., longest candidate selection can be thought to be useful assuming that the sentences in the clusters are similar to each other and the longest sentence in the cluster can be the true representative sentence.

The third method considers a sentence in a cluster as the representative sentence if it is closest to the common centroid. The common centroid is considered as the pseudo document consisting of a number of words whose weight is greater than a predefined threshold and it is the centroid of the cluster of input documents. We compute the weight of a word using the formula: log (1+tf), where tf (term frequency) = total number of times a term (word) occurs in the input collection of documents (stop words are not taken into count). Closeness of a sentence to the centroid is measured by summing up the weights of centroid words appearing in the sentence.

Assuming the above three methods as the baselines for representative sentence selection, we seek for the better solution for this. Then we proposed a new method for representative sentence selection. In this method, we calculate the importance of a sentence based on local importance and global importance of the words contained in it. The local importance of a word in a cluster indicates how much the word contributes in the formation of the central concept embodied in to a cluster and the global importance indicates how the word contributes in the formation of multiple different concepts (sub-topics) spread over the input collection of documents. The local importance of a word is calculated using log (1+CTF), where CTF (cluster term frequency) is a count of a word in the cluster. The global importance of a word is calculated using log (1+CF), where CF (cluster frequency) is the number of clusters that contain the word. Finally,

importance of a sentence is calculated using the following formulas.

$$\text{Score}(S)=\sum_{w \in S} \text{Weight}(w) \quad (4)$$

and $$\text{Weight}(w)=\alpha_1 \log(1+CTF) + \alpha_2 \log(1+CF)). \quad (5)$$

Where Score (S) indicates the importance of the sentence S and Weight (w), importance of the word w, is computed by taking weighted average of the local and global importance of the word w. In this setting, we take $\alpha_1=\alpha_2=0.5$.

After ranking sentences in the cluster based on its scores, the sentence with highest score is selected as the representative sentence.

3.5 Summary generation

After clustering the sentences, the clusters are ordered using cluster-ordering algorithm. One representative sentence from each cluster is chosen by the representative selection algorithm. We select one sentence from the top most cluster first and then continue selecting the sentences from the subsequent clusters in ordered list until a given summary length is reached.

4. SUMMARY EVALUATION

It is difficult to judge whether a generated summary is good or bad. Manual evaluation is subjective and it generally requires a lot of human efforts. Recently, the automatic evaluation measure has been popular. One such popular automatic summary evaluation method introduced in [39] is based on n-gram overlap between the system-produced and reference summaries. The concept has been implemented by ROUGE package[1] [40]. As such, the method for evaluating summary using n-gram overlap is a recall-based measure, and it requires that the summary length should controlled to allow meaningful comparisons.

We chose as our input data the document sets used in the task2 for the evaluation of multi-document summarization during the Document Understanding Conference (DUC) in 2004. This collection contains 50 test document sets, each with approximately 10 news stories. For each document set, four human-generated summaries are provided for the target length of 665 bytes (approximately 100 words).

5. EXPERIMENTS, RESULTS AND DISCUSSION

We conducted multiple experiments with the sentence-clustering based multi-document summarization. For each experiment, input data sets are preprocessed by removing stop words, but no stemming is applied. Each experiment deals with a summarization method in which sentence-clustering algorithm remains the same, but cluster ordering and representative selection techniques are replaced with the possible alternatives to judge whether summarization performance depends on the cluster ordering and representative selection techniques. For clustering ordering we tried the following two methods:

- Ordering clusters based on merely the counts of the sentences in the clusters (ClusterOdering-SC)
- Ordering clusters based on information richness (ClusterOrdering-IR)

For representative selection from clusters, we tried the following four methods:

- Select randomly a sentence from a cluster as a representative (Rep-Random)
- Select longest candidate in a cluster as a representative (Rep-Longest)
- Select a sentence closest to the centroid as a representative (Rep-Centroid)
- Select a sentence from a cluster based on local and global importance (Rep-LG)

The details of the above mentioned cluster ordering methods and representative selection methods has been discussed in the section 3. To facilitate discussion of the results of our experiments, we give short names to the different components of the system. The short names are mentioned above in brackets. For each experiment, we develop a clustering based summarization system. Each system contains three major components: similarity histogram based incremental sentence clustering (Clustering-Histo), one of cluster ordering techniques and one of representative selection techniques. We tested all these variants of the clustering based summarization systems on DUC2004 data set to find out the best among them. A summary of 665 bytes (100 words approx.) is generated for each input document set belonging to DUC2004 data collection. The fixed summary length of 665 bytes is maintained for all experimental cases due to the meaningful comparisons among them. Table 1 summarizes the performances of the various clustering based multi-document summarization systems constructed for our experiments. In table 1, row1 shows that summarizer which performs the best has the components:

[1] We retrieve ROUGEeval-1.4.2 from http://www.haydn.isi.edu/ROUGE/ in 2004

(1) histogram-based sentence clustering, (2) Information richness-based cluster ordering and (3) representative sentence selection based on local and global word importance. We consider the systems mentioned in the row2, row3, row4, row5 and row6 as the baseline sentence-clustering based summarization systems.

Table 1. Rouge-1 scores (with 95% confidence interval) for the sentence clustering based -summarization system and its variants. Summaries are stemmed before evaluation.

	ROUGE-1 score
Sentence Clustering + ClusterOrdering-IR + Rep-LG	0.3756
Sentence lustering+ClusterOrdering-IR + Rep-Centroid	0.3698
Sentence lustering+ClusterOrdering-IR+ Rep-Random	0.3684
SentenceClustering+ClusterOrdering-IR+ Rep-Longest	0.3678
Sentence Clustering + ClusterOrdering-SC + Rep-Centroid	0.3545
Sentence lustering+ClusterOrdering-SC + Rep-LG	0.3541

Row1 of the table1 shows that system (Sentence Clustering + ClusterOrdering-IR + Rep-LG), having components: sentence clustering, information-richness based cluster ordering and local-global importance based representative selection, performs better than all other variants of the sentence clustering based approaches we considered. So, we claim that only the sentence clustering is not sufficient for achieving better performance of the clustering based summarization system and it also equally depends on other two important factors: how to order clusters and how to select representatives from clusters.

During tuning the parameters of our sentence-clustering algorithm, we observe that some relaxation in the histogram ratio (that is, slight deviations from the perfect clustering) improves performance of the summarizer. The reason, which we identified, is that the perfect clustering converts the most of the clusters to singletons, which degrades the performance. Though slight relaxation in the intra-cluster cohesiveness generates a set of clusters where some clusters share some common information, it is not found unusual because same entity may participate in multiple sub-topics (or events).

Since we have tested our system on DUC 2004 data set, we have compared the ROUGE scores of our system with the official ROUGE scores of the systems participating in DUC 2004. In table 2, we have shown ROUGE scores for top five systems participated in DUC 2004 along with the system that was considered as the baseline system in DUC 2004 (indicated by peer code 2). One baseline was defined on task2 in DUC 2004. It takes the first 665 bytes of the text of the most recent document as the summary.

Table 2. Official ROUGE scores (with 95 % confidence intervals) for top five systems and one baseline system submission (peer code:2) participated in DUC 2004 Tasks 2.

Peer Code	ROUGE-1 score
65	0.3822
104	0.3744
35	0.3743
19	0.3739
124	0.3706
2(baseline)	0.32419

In table2, peer codes 65 to 124 indicate codes for the top five systems participated on task2 in DUC 2004.

Our system and its other variants mentioned in table1 perform better than the baseline (indicated by peer code 2 in table 2).

At row1 in table 1, we find that ROUGE-1 score of the proposed clustering based summarization system with the components: Histogram Based Sentence Clustering + ClusterOrdering-IR + Rep-LG is 0.3756 which is better than ROUGE-1 score of the system with peer code 104 shown at row 2 in table 2. On comparing the ROUGE scores, we find that the proposed clustering based system with the components: Histogram Based Sentence Clustering + ClusterOrdering-IR + Rep-LG, performs better than the system which was regarded as the second best (peer code 104) in DUC 2004. Though the proposed system performs slightly worse than the best system CLASSY (peer code 65 shown at row1 in table 2), the proposed system differs from the best system in many ways: the system CLASSY used Hidden Markov Model (HMM), which requires a large amount of training data. HMM states are determined based on empirical testing. In addition, CLASSY also incorporates a linguistic component as a preprocessing stage to provide the summarization engine simplified (shortened) sentences as input. So, the system, CLASSY is not easily portable to new domain and new language. In comparison to the best system (CLASSY), our system is easily portable to new domain and new language and the performance of our system is also comparable.

6. CONCLUSION

In this paper we present a sentence clustering based multi-document summarization system whose performance is comparable to the top performing multi-document summarization systems participated on task2 on DUC 2004. We also investigate on the other variants of this system. Our work focuses on the design of a successful clustering based summarization and the related issues such as how to cluster sentences, how to order clusters and how to select representative sentences from the clusters. Our experiment shows that the performance of a clustering based multi-document summarization can be made competitive with the best top performing multi-document summarization systems. To make the our system portable to new domain and new language, we did not apply stemming on the input and we did not incorporate features such as length, sentence position, cue phrase in this work though these features are proven to effective in the news domain. So, in terms of domain independency and language independency our approach is also better.

The performance of our system can be improved by improving its different components. How to measure similarity between sentences is also a crucial issue in sentence clustering based summarization approach. The better similarity measure will improve the clustering performance and this may improve the summarization performance. Incorporation of sentence simplification component at the preprocessing step of the proposed approach may improve the summarization performance while producing multi-document short summaries, because longer sentences occupies more space which prevents the other sentences from being selected in to the summary. But, the sentence simplification often requires the input to be parsed and parsing is a language dependent task.

REFERENCES

1. K.R. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, J.L. Klavans, A. Nenkova, C. Sable, B. Schiffman and S. Sigelman. Tracking and summarizing news on a daily basis with Columbia's NewsBlaster. *In Proceedings of Human Language Technology Conference (HLT 2002),* (San Diego, CA, Mar. 2002).
2. D. R. Radev, H. Jing, M. Sty, D. Tam. Centroid-based summarization of multiple documents. *Journal of Information Process and Management*. 40(6): 919-938(2004)
3. D. R Radev., H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies. *In ANLP/NAACL Workshop on Summarization*, Seattle, April, (2000).
4. J. G. Carbonell and J. Goldstein. The use of MMR, diversity-based re-ranking for reordering documents and producing summaries. *In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, (1998), pages 335–336.
5. K. McKeown, J. Klavans, V. Hatzivassiloglou, R. Barzilay and E. Eskin. Towards multi-document summarization by reformulation: Progress and prospects. *In Proceedings of the 16th National Conference of the American Association for Artificial Intelligence (AAAI-1999),* 18–22 July, pages 453– 460.
6. D. Marcu and L. Gerber. An inquiry into the nature of multi-document abstracts, extracts, and their evaluation. *In Proceedings of the NAACL-2001 Workshop on Automatic Summarization*, Pittsburgh, June. NAACL, (2001), pages 1–8.
7. E.Boros, P.B.Kantor and D.J.Neu. A Clustering Based Approach to Creating Multi-Document Summaries. *In Proceedings of the 24th ACM SIGIR Conference*, LA, 2001.
8. H. Hardy, N. Shimizu, T. Strzalkowski, L. Ting, G. B. Wise and X. Zhang. Cross-document summarization by concept classification. *In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Finland, (2002), pp. 121–128.
9. M. F. Moens, C. Uyttendaele and J. Dumortier. Abstracting of legal cases: the potential of clustering based on the selection of representative objects. *Journal of the American Society for Information Science*, 50 (2), (1999), 151-161.
10. K.M. Hammouda and M.S. Kamel. Efficient Phrase-Based Document Indexing for Web Document Clustering. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 10, (2004), 1279-1296.
11. P. B. Baxendale. Man-made index for technical literature—An experiment. *IBM Journal of Research and Development*, (1958), 2(4): 354–361.
12. H. P. Edmundson. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, (1969), 16(2): 264–285.
13. Luhn, H. P. The automatic creation of literature abstracts. *IBM Journal of Research Development*, 2(2), (1958), 159–165.
14. K.R. McKeown and D. R. Radev. Generating summaries of multiple news articles. *In Proceedings of the 18th Annual International ACM SIGIR Conference on Re-search and Development in Information Retrieval,* Seattle, (1995), pages 74–82.
15. G. C. Stein, A. Bagga and G. B. Wise, Multi-Document Summarization: Methodologies and Evaluations. *In Conference TALN 2000*, Lausanne, (2000).
16. V. Hatzivassiloglou, J. Klavans, and E. Eskin. Detecting test similarity over short passages: Exploring linguistic feature combinations via machine learning. *In Proceedings of EMNLP,* (1999)
17. V. Hatzivassiloglou, J. L. Klavans, M. L. Holcombe, R. Barzilay, M-Y. Kan, and K. R. McKeown. SimFinder: A Flexible Clustering Tool for Summarization. *NAACL, Workshop on Automatic Summarization*. Pittsburgh, PA, (2001).
18. W. Cohen. Learning trees and rules with set-valued features. *In Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI), (*1996).

19. R. Barzilay, Elhadad, and K. McKeown. Sentence ordering in multi-document summarization. *In Proceedings of the Human Language Technology Conference* (2001).
20. M Witbrock and V. O. Mittal. Ultra-Summarization: A Statistical Approach to Generating Highly Condensed Non-Extractive Summaries. *In SIGIR99, (1999),* pp. 315-316 Berkeley, CA.
21. H. Jing. Using hidden Markov modeling to decompose human-written summaries. *Computational Linguistics*, (2002), 28(4), 527–543.
22. K. Knight and D. Marcu. Statistics-based summarization | step one: Sentence compression. *In Proceeding of the 17th National Conference of the American Association for Artificial Intelligence (AAAI-2000),* (2000), pp. 703-710.
23. D. R.. Radev and K. R. McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics,* (1998), 24 (3), 469-500.
24. R. Barzilay, K. McKeown, and M. Elhadad. Information fusion in the context of multi-document summarization. *In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, College Park, MD, 20–26 June, (1999), pages 550–557.
25. J. Conroy, J. Schlesinger, J. Goldstein,and D. O'Leary. Left-brain/right-brain multi-document summarization. *In Proc. of DUC*, (2004).
26. X. Wan and J. Yang. Improved affinity graph based multi-document summarization. *In Proceedings of HLT-NAACL, Companion Volume: Short Papers*, (2006), pages 181–184..
27. I. Mani, B. Gates and E. Bloedorn. Improving summaries by revising them. *In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL 99),* College Park, MD, June, (1999), pages 558–565.
28. C. Lin. Improving Summarization Performance by Sentence Compression- A Pilot Study. *In the Proceedings of the Sixth International Workshop on Information Retrieval with Asian Language (IRAL 2003)*, Sapporo, Japan, (2003).
29. K. Knight and D. Marcu, 2000. Statistics-Based Summarization-Step One: Sentence Compression. In Proceedings of AAAI, Austin, TX, USA, (2000).
30. E. Hovy, C. Lin and L. Zhou. A BE-based Multi-document summarizer with sentence compression. In Proceedings of Multilingual Summarization Evaluation (ACL 2005 workshop), Ann Arbor, MI, (2005).
31. I. Mani and E. Bloedorn. Multi-document summarization by graph search and matching. *In Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97),* Providence, Rhode Island., American Association for Artificial Intelligence, (1997), pp. 622-628
32. R. Barzilay and M. Elhadad Using Lexical Chains for Text Summarization. In Mani, I., & Maybury, M. T. (Eds.), *Advances in Automatic Text Summarization*, The MIT Press, (1999), pp. 111-121.
33. J. Kupiec, J. O. Pedersen and F. Chen. A trainable document summarizer. In Research and Development in Information Retrieval, (1995)., pp. 68-73.
34. C.-Y. Lin,. Training a Selection Function for Extraction. *In Proceedings of the Eighteenth Annual International ACM Conference on Information and Knowledge Management (CIKM),* ACM , Kansas City. (1999), pp. 55-62.
35. M. Osborne. Using Maximum Entropy for Sentence Extraction. *In ACL Workshop on Text Summarization*, (2002).
36. H. Daume' III and D. Marcu. A phrase-based HMM approach to document/abstract alignment. *In Proceedings of EMNLP,* Barcelona, Spain. Association for Computational Linguistics, (2004), pp. 119-126.
37. G. Erkan and D. R. Radev. LexRank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*, (2004).
38. X. Wan: Using only cross-document relationships for both generic and topic-focused multi-document summarizations. *Information Retrieval (2008)* 11:25–49
39. C.-Y. Lin. and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence. *In Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003)*, Canada, May 27 - June 1, (2003).
40. C.Y. Lin. 2004b. ROUGE: A package for automatic evaluation of summaries. *In WAS 2004: Proceedings of the Workshop on Text Summarization Branches Out*, July 25–26, 2004, Barcelona, Spain.