Centroid Based Summarization of Multiple Documents Implemented using Timestamps

Prof. R. Nedunchelian
Professor and Head, Department of computer science and engineering
Sri Venkateswara college of engineering
Email:nedun@svce.ac.in

ABSTRACT

We propose a multiple-document summarization system with user interaction. We introduce a system that would extract a summary from multiple documents based on the document cluster centroids, which is effectively the distribution of terms in the multiple documents in the cluster. This summarization technique is a cluster-based, extractive summarization method, where passages are first clustered based on similarity, prior to the selection of passages that form the extractive summary of the documents. The sentences are then issued a timestamp based on the order of their occurrence in the original document, thereby ensuring the chronological order of sentences. Passage clustering forms a main component in this system that aims to extract the most relevant sentences of the documents at the same time keeping the summary non-redundant. The implementation is based on the MEAD extraction algorithm and redundancy based algorithm. MEAD extraction algorithm uses three features to compute the salience of the sentence. They are Centroid value, Positional value and First-sentence overlap. Redundancy algorithm checks for overlapping words in sentences and issues a redundancy penalty. Timestamps are issued to sentences to maintain the chronological order of the sentences and hence a coherent and free-flowing summary can be generated.

1. Introduction

The world of text is huge and expanding. As illustrated by the World Wide Web (WWW), important information will continue to be available as text despite the growing use of multi media content. A large percentage of the data available electronically is in the form of natural language text. The information contained in natural language text is highly unstructured unlike databases which are highly structured. Retrieving useful information from natural language text offers its own set of challenges due to this inherently ambiguous nature of natural language text. Information overload is another issue that affects the user due to the vast amount of information available often from different sources with different and often contradictory perspectives. Today, it is more difficult identifying and spotting what one requires from the vast data source available and it would be infinitely simpler for the user to have a summary of the information available to him, taking various viewpoints into consideration, and at the same time not overburdening him with too much information.

2. Automatic Summarization

Automatic summarization is the process of taking information source, extracting content from it, and presenting the most important content of the user in a condensed form and in a manner sensitive to the user's or application's need. It does not start with a predefined set of criteria of interest. Based on the type of the input text the summary generators are classified as single or multi document summary generators.

3. Motivation

Recent rapid progress of computer and communication technologies enabled us to access enormous amount of machine-readable information easily. However, this has caused the information overload problem. In order to solve this problem multiple-document summarization has been increasingly used and the multiple-document summarization technology has been intensively studied recently. Imagine that a reader is using an internet search engine, searching for certain topics of interest over the internet. The information retrieval system returns a long list of html documents, possibly related. Without having to check these links one by one, is there a possible way to generate a unified summary of the information contained in these documents? Here is another scenario: online news services cluster news stories from different news agencies and present them to the reader. The different news stories in one cluster, presumably on the same topic, have major

978-0-7695-3267-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ICETET.2008.122

IEEE computer society

overlaps in their contents, while some of them have certain sides of the topic unique to the rest of the news stories. Is it possible to generate a single, preferably short, article giving the reader a consolidated story concerning this news topic? The above are two typical situations when the information contents in different documents need to be summarized and synthesized. The fact that information is stored in a distributed and diversified fashion has hence created an acute need for multi-document summarization systems. Such systems have great potential in facilitating information processing: not only can the readers benefit from short and informative summaries, but applications can include information retrieval, information extraction, and translingual information processing as well.

4. Single v/s Multi Document Summarization

Before we start to look at multi-document summarization systems, we would like to first take a look at single document summarization systems and contrast the differences between the two tasks. There are two major differences between single and multiple document summarizations. First, most approaches to single document summarization involve extracting sentences from the document (Paice, 1990; Kupeic et al., 1995; Marcu, 1998). Indeed, sentence extraction is one particular approach to single document summarization. Given multiple documents, however, the information stored in different documents inevitably overlaps with each other. Hence effective methods that merge information stored in different documents and if possible, contrast their differences are highly desired in the case of multi-document summarization.This would usually mean that certain operations need to be taken below the sentence level, possibly involving merging, compressing or splitting sentences. Second, most single document summarization systems, to a certain extent, make use of the monolithic structure of the document. The way sentences are ordered within a document usually represents certain logic relations between the sentences. Such relations, in many cases, are usually exploited by single document summarization systems. For example, one simple but quite effective way to write a summary for a single document is to take the first sentence from each paragraph and put them together in their original order. However, for multi-document summarizations, the structure of a single document cannot be readily used in such a straightforward fashion. In this sense, multi-document summarization systems usually rely less on the structures of the documents.

5. General Overview of Our Approach

To generate a summary, one must first start with relevant documents that one wishes to summarize. The process of identifying all articles on an emerging event is called Topic Detection and Tracking (TDT). A large body of research in TDT has been created over the past years (Allan, Papka, & Lavrenko, 1998). We will present an extension of our own research on TDT (Radev, Hatzivassiloglou, & McKeown, 1999) that we used in our summarization of multi-document clusters. The main concept we used to identify documents in TDT is also used to rank sentences for our summarizer.

5.1. Topic Detection and Tracking (TDT)

Our entry in the official TDT evaluation, uses modified TF * IDF to produce clusters of news articles on the same event ('TF' indicates how many times a word appears in a document while IDF measures what percentage of all documents in a collection contain a given word). An incoming document is grouped into an existing cluster, if the TF * IDF of the new document is close to the centroid of the cluster. A centroid is a group of words that statistically represent a cluster of documents. The idea of a centroid is described further in other chapters. In our experiment we assume that an event cluster is already produced from the corpus and hence we deal with only related documents. Fro the same reason TF * IDF is also used.From a TDT system, an event cluster can be produced. An event cluster consists of chronologically ordered news articles from multiple sources. These articles describe an event as it develops over time. In our experiments, event clusters range from 2 to 10 documents. It is from these documents that summaries can be produced.

5.2. Centroid-Based Summarization (CBS)

The technique we used for our multi-document summarization is centroid-based summarization (CBS). CBS uses the centroids of the clusters produced by TDT to identify sentences central to the topic of the entire cluster. We have implemented CBS in MEAD, our publicly available multi-document summarizer. A key feature of MEAD is its use of cluster centroids, which consist of words which are central not only to one article in a cluster, but to all the articles.

6. MEAD

MEAD is significantly different from previous work on multi-document summarization (Carbonell & Goldstein, 1998; Mani & Bloedorn, 2000; McKeown, Klavans, Hatzivassiloglou, Barzilay, & Eskin, 1999; Radev & McKeown, 1998), which uses techniques such as graph matching, maximal marginal relevance, or language generation.MEAD is a sentence level extractive summarizer that takes document clusters as input. Documents are represented using term frequency-inverse document frequency (TF * IDF) of scores of words. Term frequency used in this context is the average number of occurrences (per document) over the cluster. IDF value is computed based on the entire corpus. The summarizer takes already clustered documents as input. Each cluster is considered a theme. The theme is represented by words with top ranking term frequency, inverse document frequency (TF-IDF) scores in that cluster. Sentence selection is based on similarity of the sentences to the theme of the cluster (Ci).The next factor that is considered for sentence selection is the location of the sentence in the document (Li). In the context of newswire articles, the closer to the beginning a sentence appears, the higher its weightage for inclusion in summary. The last factor that increases the score of a sentence is its similarity to the first sentence in the document to which it belongs (Fi).Finally, evaluation of multi-document summaries is a difficult problem. Currently, there is no widely accepted evaluation scheme. We propose a utility-based evaluation scheme, which can be used to evaluate both single-document and multi-document summaries. The main contributions of this paper are: the use of cluster-based relative utility (CBRU) and cross-sentence informational subsumption (CSIS) for both single and multi-document summaries, the development of a centroid-based multi-document summarizer.

7. A Generalized Framework for Multi-Document Summarization

Given the input as a cluster of documents on the same topic, the task of a multi-document summarization system is to generate a short paragraph that preserves the majority of information contained in the original documents. In reality, this process is seldom done in a single step. Rather, it can be thought of as a multistage compression process, described as follows:

Step 1:
The system first takes the set of documents and breaks them into sentences as input.
Step 2:
The sentences are clustered into sentence groups. (This step can be optional.)
Step 3:
For each sentence group, one sentence level representation is generated or chosen.
Step 4:
The sentence level representation is either generated as a linear sentence, or further compressed if necessary.

The above four stage compression process takes the documents as the input, and at each stage, reduces the complexity in the representation using various techniques. The following graph gives a high level view of a multi-document summarization system.

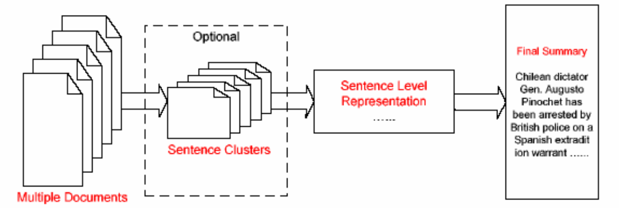
Optional
Multiple Documents
Sentence Clusters
Sentence Level Representation
……
Final Summary
Chilean dictator Gen. Augusto Pinochet has been arrested by British police on a Spanish extradit ion warrant ……

Figure 1. A generalized framework of multi-document summarization

Most multi-document summarization systems can be viewed as an instance of this framework. For a multi-document summarization system to function, multiple components would be needed, each of which reduces the previous representation in terms of complexity or length and feed it to the next stage component. Each of these components would involve techniques such as clustering, classification, information fusion, and/or transduction based compression. The detailed discussions of such techniques are to be covered in the later sections of this project.

8. Topic Detection and Tracking

The TDT is intended to explore techniques for detecting the appearance of new topics and for tracking the reappearance and evolution of them. During the first portion of this, the notion of a “topic” was modified and sharpened to be an “event”, meaning some unique thing that happens at some point in time. The notion of an event differs from a broader category of events both in spatial/temporal localization and in specificity. For example, the eruption of Mount Pinatubo on June 15th, 1991 is considered to be an event, whereas volcanic eruption in general is considered to be a class of events. Events might be unexpected, such as the eruption of a volcano, or expected, such as a political election.
The TDT assumes multiple sources of information, for example various newswires and news broadcast programs.

The information flowing from each source is assumed to be divided into a sequence of stories, which may provide information on one or more events. The general task is to identify the events being discussed in these stories, in terms of the stories that discuss them. Stories that discuss unexpected events will of course follow the event, whereas stories on expected events can both precede and follow the event.

The Topic Detection and Tracking is concerned with the detection and tracking of events. The input to this process is a stream of stories. This stream may or may not be pre-segmented into stories, and the events may or may not be known to the system (i.e., the system may or may not be trained to recognize specific events). This leads to the definition of three technical tasks to be addressed in the TDT. These are namely the tracking of known events, the detection of unknown events, and the segmentation of a news source into stories.

8.1. Segmentation Task

The segmentation task is defined to be the task of segmenting a continuous stream of text (including transcribed speech) into its constituent stories. To support this task the story texts from the study corpus will be concatenated and used as input to a segmentor. This concatenated text stream will include only the actual story texts and will exclude external and internal tag information. The segmentation task is to correctly locate the boundaries between adjacent stories, for all stories in the corpus.

9. Informational Content of Sentences

9.1. Cluster-Based Relative Utility (CBRU)

Cluster-based relative utility (CBRU, or relative utility, RU in short) refers to the degree of relevance (from 0 to 10) of a particular sentence to the general topic of the entire cluster (for a discussion of what is a topic, see Allan, Carbonell, Doddington, Yamron, & Yang, 1998). A utility of 0 means that the sentence is not relevant to the cluster and a 10 marks an essential sentence. Evaluation systems could be built based on RU and thus provide a more quantifiable measure of sentences.

9.2. Cross-Sentence Informational Subsumption (CSIS)

A related notion to RU is cross-sentence informational subsumption (CSIS, or subsumption). CSIS reflects that certain sentences repeat some of the information present in other sentences and may, therefore, be omitted during summarization. If the information content of sentence **a** (denoted as **i(a)**) is contained within sentence **b**, then **a** becomes informationally redundant and the content of **b** is said to subsume that of **a**:

$$\mathbf{i(a)<i(b)}$$

In the example below, (2) subsumes (1) because the crucial information in (1) is also included in (2) which presents additional content: ‘‘the court’’, ‘‘last August’’, and ‘‘sentenced him to life’’.
(1) John Doe was found guilty of the murder.
(2) The court found John Doe guilty of the murder of Jane Doe last August and sentenced him to life.

10. Mead Extraction Algorithm

MEAD is a publicly available toolkit for multi-lingual summarization. The toolkit implements multiple summarization algorithms (at arbitrary compression rates) such as position-based, Centroid[RJB00], TF*IDF, and query-based methods.MEAD v1.0 and v2.0 were developed at the University of Michigan in 2000 and early 2001.

10.1. Mead Functionality

MEAD can perform many different summarization tasks. It can summarize individual documents or clusters of related documents (multi-document summarization).
MEAD includes two baseline summarizers: lead-based and random. Lead-based summaries are produced by selecting the first sentence of each document, then the second sentence of each, etc. until the desired summary size is met. A random summary consists of enough randomly selected sentences (from the cluster) to produce a summary of the desired size. MEAD is a sentence level extractive summarizer that takes document clusters as input. Documents are represented using term frequency-inverse document frequency (TF-IDF) scores of words. Term frequency used in this context is the average number of occurrences (per document) over the cluster. IDF value is computed based on the entire corpus. The summarizer takes already clustered documents as input. The cluster is represented by words with top ranking term frequency, inverse document frequency (TF-IDF) scores in that cluster.

10.2. Criteria

The philosophy of MEAD is based on optimizing the selected sentences for the following two criteria

- Cluster-based relative utility (CBRU) refers to the degree of relevance of a particular sentence to the general topic of the entire cluster.
- Cross-sentence informational subsumption (CSIS) reflects that sentences repeat some of the information present in other sentences and may be omitted during summarization.

The multi-document summarizer should ideally maximize CBRU and minimize CSIS at the same time. Before being sent to MEAD as the input, the documents, if not already clustered, are clustered into document clusters using centroid-based clustering techniques

MEAD represents each document cluster as a long vector, each component of which is a word appeared in the cluster. Let *D* be the document cluster r and let |*D*| be the number of documents in the document cluster. This vector, defined as the centroid of the cluster, is given as follows:

$$\text{Centroid}(D) = (v_{w1}, v_{w2}, \ldots, v_{wn}),$$
$$\text{where } v_{wi} = [TF(w_i)\ IDF(w_i)] / |D|$$

The stop words are removed from centroid representation by including these words in the rejrection list which will not be considered for calculation of centroid of the sentences. Thus the words are removed from the centroid representation. Let $S_{i,k}$ denote the ith sentence in the document D_k belonging to D. Then we define three features for choosing the sentences:

10.3. Centroid value

The centroid value for sentence $S_{i,k}$ is defined as the normalized sum of the centroid components:

$C'_{i,k} = \Sigma_{w \in Si,k}$ (([TF(w_i) IDF(w_i)]/ |*D*|)

which is normalized sum of the centroid values.

For example, the sentence ‘‘President Kalam met with Manmohan Singh in January’’ would get a score of 243.34 which is the sum of the individual centroid values of the words (Kalam =36.39; Manmohan =47.54; Singh =75.81; January =83.60).

10.4. Positional Value

For every sentence *i,k S* , suppose it is coming from document D_k, where length(n)= D_k is the number of sentences in D_k . The positional value for this sentence is computed as:

$P_{i,k}$ = ((n-i+1)/n)* C_{max}

10.5. First Sentence Overlap

The overlap value is computed as the inner product of the sentence vectors for the current sentence i and the first sentence of the document. The sentence vectors are the n-dimensional representations of the words in each sentence, whereby the value at position i of a sentence vector indicates the number of occurrences of that word in the sentence.

$$\text{Fi,k} \quad = \quad \underline{S_{i,k}} \bullet \underline{S_{1,k}}$$

The system then defines the raw score for each sentence.

SCORE (S_i) = $w_c C_{i,k} + w_P P_{i,k} + w_F F_{i,k}$

w_c , w_P , w_F are weights of the features.

11. Short Comings of MEAD

The summary produced by MEAD contains the selected sentences from each document and output them in the order prevalent in the original document. Hence the sentences selected from the first document will appear before the sentences selected from the second document , similarly selected sentences from the second document will appear before the sentences selected from the third document and subsequently. Thus the order of the sentences in the summary may not be logical in occurrence. Hence to overcome this short coming we have implemented a concept called timestamps in our work.

Table 1. A Sample Score Table

Document	Sentence	Timestamps	Score
2	1	1	62.31
1	4	4	60.24
1	3	3	56.21
3	6	6	54.3
2	2	2	41.23
1	5	5	36
2	6	6	26.12
3	2	2	22.5

11.1 Timestamps

The implementation of timestamps is carried out by assigning a value to each sentence of the document depending on the chronological position in which it occurs in the document. Once the sentences are selected they are arranged in the ascending order depending on the timestamps. This gives the summary an ordered look, bringing out a coherent looking summary. The number of sentences in the summary is dictated by the compression rate. For example if the compression rate is 10% and the total number of sentences in all documents is equal to 100, then there will be 10 sentences in the summary. Consider the following example. Let there be three documents. Assume that the following sentences from the various documents have been selected to form the summary depending on the compression rate and score values. These sentences will be displayed in the following order if it is produced by Centroid based summarization approach as given in the Table 2. Thus we can see the absence of logical ordering in the summary because the order of occurring of events is not maintained. These sentences will be displayed in the following way if produced in our work.

Table 2. Output Produced By MEAD

Document	Sentence	Timestamps	Score
1	4	4	60.24
1	3	3	56.21
1	5	5	36
2	1	1	62.31
2	2	2	41.23
2	6	6	26.12
3	6	6	54.3
3	2	2	22.5

Table 3. Output when Timestamp is Implemented

Document	Sentence	Timestamps	Score
2	1	1	62.31
2	2	2	41.23
3	2	2	22.5
1	3	3	56.21
1	4	4	60.24
1	5	5	36
3	6	6	54.3
2	6	6	26.12

Thus using timestamps the information conveyed in the summary remains the same but is presented in a more logical manner maintaining the chronological occurrence of the events.

12. Conclusion and Future work

Automatic text summarization deals with condensing the content of an information source and presenting it in a manner sensitive to the user's needs. An automatic text summarizer was created, taking as input multiple documents, all dealing with the same topic, as input, scoring the various sentences, and then using timestamps to generate the summary. Word scoring was done using word frequency as a parameter. Sentence scoring consisted of the sum of the word scores in the sentence and also took as parameters, the absolute location of the sentence in the document and the relative position within a paragraph. After scoring was done, an evolutionary approach was used by issuing a timestamp to every sentence, a timestamp is a value assigned to every sentence so as to maintain the chronological order of the sentences in the summary and hence the resultant summary is a logical, coherent and concise form of the given input documents with redundant sentences removed. Another useful feature is that the length of the summary can be adapted to the user's needs as can the number of articles to be summarized. The compression rate can be specified by the user so that he can choose the amount of information he wants to imbibe from the documents. A possible application of our work can be made to make data available on the move on a mobile network by even shortening the sentences produced by our algorithm and then shortening it. Various NLP based algorithms can be used to achieve this. Thus we would first produce a summary by sentence extraction from various documents and then abstractive methods are employed to shorten those sentences produced. This will ensure that the summary produced is to the highest condensed form which can be made of in the mobile industry.

13. References

[1] Carbonell, J. G., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In A. Mof-fat, & J. Zobel (Eds.), Proceedings of the 21st annual international AC M SIGIR conference on research and development in information retrieval (pp. 335–336). Melbourne, Australia.

[2] Dragomir R. Radev, Hongyan Jing, Malgorzata Stys, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, 2004.

[3] Goldstein, J., Kantrowitz, M., Mittal, V. O., & Carbonell, J. G. (1999). Summarizing text documents: sentence selection and evaluation metrics. In Research and development in information retrieval (pp. 121–128). Berkeley, California.

[4] Jacques Robin. 1994. Revision-based generation of natural language summaries providing historical background: corpus based analysis, design, implementation and evaluation. Ph.D. thesis, Department of Computer Science, Columbia University, NY.

[5] Kathleen McKeown, Judith Klavans, Vasilis Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: progress and prospects, in *Proceedings of AAAI*, 1999, Orlando, Florida.

[6] Madhavi K. Ganapathiraju , 2002 , Relevance of Cluster size in MMR based Summarizer: A Report,

[7] Radev, D. R., Hatzivassiloglou, V., & McKeown, K. R. (1999). A description of the CIDR system as used for TDT-2. In DARPA broadcast news workshop. Herndon, Virginia.

[8] Radev, D. R., & Tam, D. (2003). Single-document and multi-document summary evaluation via relative utility. In Poster Session, Proceedings of the ACM CIKM conference. New Orleans, LA.

[9]Radev, Allison, Blair-Goldensohn et al (2004), MEAD - a platform for multidocument multilingual text summarization