Approaches To Summarize Multi Documents Using Information Extraction

Hari Om Sharan, Rajeev Kumar, Garima Singh, Mohammad Haroon

Department of Computer Science, College of Engineering, Teerthanker Mahaveer University, Moradabad (U.P.), India

Email: rajeevphd@hotmail.com, sharan.hariom@gmail.com, haroonayme@gmail.com

Abstract: No one has time to read everything, yet we often have to make critical decisions based on what we are able to understand. With summaries, we can make effective decisions in less time. Thus the technology of automatic text summarization is becoming essential to deal with the problem of information overload. Text summarization is the process of extracting the most important information from a single document or from a set of documents to produce a short and information rich summary for a particular user or task. Multi-document summarization is an automatic procedure for extraction of information from multiple texts about the same topic. Most of the MDS systems have been based on an extraction method, which identifies key textual segments (e.g., sentences or paragraphs) in source documents and selects them for the summary. Multi-document summarization aims to distill the most important information from a set of documents to generate a compressed summary. In this paper we are introducing various approaches for multi document summarization using information extraction.
[Hari Om Sharan, Rajeev Kumar, Garima Singh, Mohammad Haroon. Approaches To Summarize Multi Documents Using Information Extraction. Academia Arena. 2011;3(7):62-67] (ISSN 1553-992X). http://www.sciencepub.net.

Keywords: Summarize; Multi Document; Information Extraction

Introduction

In the literature, the summaries are considered of two types: Extractive summaries and Abstractive summaries. An extractive summary is generated by selecting sentences from the document(s), while an abstractive summary can have non-existing words or sentences in the original document(s). In addition to the initial research area of single-document summarization, recent research work has focused on multi-document summaries. In multi-document summarization, the generated summary is created by the synthesis of multiple documents instead of a single document.

The aim of the multi-document summarization is to present multiple documents in form of a short summary. This short summary can be used as a replacement for the original documents to reduce, for instance, the time a reader would spend if she/he were to read the original documents. Various approaches have been applied in multi-document summarization task. Few of the important approaches for MDS using information extraction are described in this paper.

The goal of the automatic text summarization is to provide a user with a presentation of the substance of a body of material in a coherent and concise form to save time and effort. Ideally, a summary should contain only the “right” amount of the interesting information and it should omit all the redundant and “uninteresting” material. The summary produced by automatic summarization can be of two types-generic or user specific. The generic summaries contain the over all most salient information from the original documents while the user specific summaries contain the most relevant information depending upon the choice and interests of the user.

Automatic text summarization can be broadly categorized in two types based on the number of source documents: Single Document Summarization and Multi Document Summarization (MDS).

Single Document Summarization

As the name suggests in single document summarization there is only one large source document Single document summarization is easy as compared to multi document summarization task. As in single document summarization there is no issue of multiple languages, multiple input format, writing style, redundancy of information etc.

Multi-document Summarization

In case of multi document summarization the information is distributed over multiple source documents. The multi-document summarization task has turned out to be much more complex than summarizing a single document, even a very large one. This difficulty arises from inevitable thematic diversity within a large set of documents. These documents can be in different languages, written by different authors having different background knowledge and different document formats. A good summarization technology aims to combine the main themes with completeness, readability, and conciseness. An ideal multi-document summarization system does not simply shorten the source texts but presents information organized around the key aspects to represent a wider diversity of views on the topic. When such quality is achieved, an automatic multi-document summary is perceived more like an over view of a given topic.

The multi document summarization can be categorized along two different dimensions: abstract-based [2, 3] and extract-based [4, 5]. An extract-summary consists of sentences extracted from the document while an abstract-summary may employ words and phrases that do not appear in the original document. The extractive summarization tries to select a number of indicative sentences, passages or paragraphs from the original document according to a target summarization ratio, and then sequence them together to form summary. The abstractive summarization, on the other hand, tries to produce a concise abstract of desired length that can reflect the key concepts of the document. The latter seems to be more difficult, and most of the recent approaches have focused more on the extraction based summarization.

Information Extraction Approaches

There are several ways in which one can characterize different approaches to information extraction based summarization. One useful way is to examine the level of processing. Based on this, summarization can be characterized as approaching the problem at the surface, entity, or discourse levels [1].

Surface-level approaches [4, 5, 7] tend to represent information in terms of shallow features which are then selectively combined together to yield a salience function used to extract information. These features include frequency, location, background, cue words and phrases. Entity-level approaches [8, 9] build an internal representation for text, modeling text entities and their relationships. These approaches tend to represent connectivity in the text to help determine what is salient. Relationships between entities include similarity, proximity, co-occurrence, thesaural relationships among words (synonymy, antonym, parts-of relations), logical relations (agreement, contradiction, and consistency) syntactic relations. Discourse-level approaches [6, 10] model the global structure of the text, and its relation to communicative goals. This structure can include format of the document, threads of topics as they are revealed in the text, and rhetorical structure of the text, such as argumentation or narrative structure. These are the primary examples of the approaches, and many systems adopt a hybrid approach (e.g., taking a discourse level approach where the smallest segments are surface strings or entities). K.Ramanathan et.al [13] has proposed a new language independent single-document summarization method. They map document sentences to semantic concepts in Wikipedia and select sentences for the summary based on the frequency of the mapped-to concepts.

Different MDS systems use different measures in assigning the salience score to the sentences. Based on the methods the MDS systems employ in assigning salience score to the sentences, they can broadly be classified into three categories as centroid based, clustering based and graph based summarization. Here we briefly describe the general methods employed in assigning salience scores for the sentences in each of these three categories.

Clustering Based MDS

One of the first and very popular approaches to MDS was cluster topically related sentences from the input and select one sentence from the cluster as a representative of the topic in the summary [11].These summarizers obviously try to exploit frequency on the sentence level, clusters with more sentences considered more important. Again, a hidden parameter can change the results considerably since if lower similarity between sentences in the cluster is required, bigger clusters can be formed, but the sentences in them will not be tightly related on the same topic. Such an approach assigning importance to sentences also deals directly with the problem of duplication removal:

Since, only one sentence per cluster is chosen, the summary would not include repetition.

Interestingly the size of the cluster (equivalent to sentence frequency), did not lead to good information extraction performance. The problem was addressed by adding in the weighting of term frequency (TF) and inverse document frequency (IDF). The addition of such information, which incorporates in the cluster score, the frequency also of the words in the sentences, leads to much better results in information extraction.

Centroid Based MDS

The centroid-based method is one of the most popular extractive summarization methods. MEAD is an implementation of the centroid-based method. Radev et.al. [7] described an extractive multi document summarizer (MEAD) which chooses a subset of sentences from the original documents based on the centroid of the input documents. For each sentence in a cluster of related documents, MEAD computes three features and uses a linear combination of the three to determine the most important sentences. The three features used are centroid score, position, and overlap with first sentence (or the title).

- The centroid score C_i is a measure of the centrality of a sentence to the over all topic of a cluster.
- The position score P_i which decreases linearly as sentence gets farther from the beginning of the document.

- The overlap with first sentence score Fi which is the inner product of the tf-idf weighted vector representations of a given sentence and the first sentence (or title) of the document. All three features are normalized (0-1) and the over all score for a sentence Si is calculated as

$$W\ (Si) = Wc *Ci+Wp*Pi+W_f*Fi, \quad (1)$$

Where Wc, Wp, and W_f are the individual weight age given to each type of features respectively. Now the sentences are ranked according to their combined score which is a linear combination of all the sentence features used. All three features are normalized in the range 0–1.[14]

MEAD uses a cosine similarity metric to compare each candidate sentence (for inclusion in the summary) to each higher-ranking sentence. If the candidate sentence is too similar to the specified threshold [14], it is penalized and is not included in the summary. Finally, the top remaining n-percent of the sentences (with the compression rate 'n' being determined by the user), are returned to the user as the summary.

MEAD discards sentences that are too similar to other sentences. Any sentence that is not discarded due to high similarity and which gets a high score is included in the summary.

How the output sentences are ordered is an issue with MEAD. Timestamps are not always available given a set of documents. Sometimes, the sorting of the Timestamps can reach a tie. When a tie occurs, if both the last sentence of one document and the first sentence of the other document are chosen, MEAD can potentially put the former right before the latter in the final summary, which may bring questionable results with regards to the inter-sentence logic.

In general, MEAD is not a trained system. Although Radev et al. (2004) suggested that a training set can be used, the features that such a training process can use are only three: centroid, positional and first sentence overlap. Trainable summarization system was proposed as early as (Kupeic et al., 1995) and recently in (Barzilay and Lee, 2004). It would be interesting to see how a richer feature set would affect the system performance.

Term Frequency Based MDS

Most of the extraction based multi-document summarization systems take advantage of the frequency of individual words. The more number of times a word occur in the source documents increase the chances of it to be included in the summary. The term frequency is the prime feature in summarization for the TF- IDF based multi-document summarization systems [12]. Here TF represents the term frequency that is the frequency of a word in a document, and IDF represents the inverted document frequency that is the distribution of a term in the whole corpus of data and is equal to the number of documents which contains the term divided by total number of documents in the corpus.

The content that appears frequently in the input has a higher likelihood of being selected a human summarizer for inclusion in a summary. It is observed that high frequency words from input are very likely to appear in the human summary. This confirms that unigram (individual word) frequency is one of most important the feature that impact a human's decision to include specific content in a summary. But the co-occurrence of the individual words in the inputs and the human summaries does not necessarily entail that the same facts have been covered. A better granularity for such investigation is the sequence of such individual words, such as the summary sentences. Thus the overlapping of a sequence summary confirms that both the documents contain same information. Almost all of the systems have used the unigram frequency for assigning salience scores none has selected the frequency of more than single words which conveys more meaning for the assignment of salience score.

One formal method to capture this phenomenon would model the appearance of words in the summary under a multinomial distribution. That is, for each word w in the input vocabulary, we associate a probability p (w) for it to be emitted into a summary. The overall probability of the summary then is

$$(1) \quad \frac{N!}{n_1!\ldots n_r!} \quad p(w_1)^{n}{}_{1}\ldots.(w_r)^{n}{}_{r}$$

Where N is the number of words in the summary, $n_1 + \ldots + n_r = N$ and for each i, n_i is the number of times word w_i appears in the summary and $p(w_i)$ is the probability that w_i appears in the summary.

The following algorithm provides the basis for summarization.

Step 1 Compute the probability distribution over the words w_i appearing in the input, $p(w_i)$ for every i; $p(w_i) = n/N$, where n is the number of times the word appeared in the input, and N is the total number of content word tokens in the input.

Step 2 For each sentence Sj in the input, assign a weight equal to the average probability of the words in the sentence, i.e.

$$\text{Weight (Sj)} =\sum wj\epsilon Sj \quad p\,(wj) \quad \overline{\qquad\qquad} \qquad |\{wj|wj\epsilon Sj\}| \qquad (2)$$

Step 3 Pick the best scoring sentence that contains the highest probability word.

Step 4 For each word wi in the sentence chosen at step 3, update their probability

$$p_{new}(wi) = p_{old}(w_i) \cdot p_{old}(w_i)$$

Step 5 If the desired summary length has not been reached, go back to Step 2.

Graph Based MDS

Some of the most newly developed summarizers are those that reduce the problem of summarization to graph problems, notably using the Page-Rank algorithm. Of these, the most successful application to multi document summarization was that of Erkanand Radev [10]. In their Lex Rank algorithm, each sentence defines a node in the text graph. To define edges in the graph, the cosine similarity between two sentences is computed and an edge is added between the nodes representing the two sentences if the similarity exceeds a predetermined threshold. Thus the edges are defined for sentences that share the same words. The Page-Rank algorithm is then used iteratively to compute the rank of each sentence as a function of the number of neighbors and the importance of the neighbors of each node. The iterations distribute the weight across the graph, and quickly explain that the iterative spreading of importance in the graph is similar to voting process:
Sentences from the entire graph vote for the sentences with which they share word overlap. Of course, such a voting procedure can be achieved by a direct frequency count, rather than distributing information little by little through the nodes. So the Page-Rank algorithm can be seen as a complex (unobservable) function that assigns weights to sentences based on the frequency of words that appear in the text. In order to avoid repetition, sentences that are assigned high importance, but are similar to more important sentences are not included in the summary.

In this section we discuss four graph based methods. They are

(i) Cumulative Sun
(ii) Degree of Centrality proposed by Erkan et al [15].
(iii) LexRank & Continuous LexRank methods and
(iv) Discounting Method

Cumulative Sum Method

In this method, any sentence weight is obtained by adding all the entries in the similarity matrix, corresponding to the sentence, either row wise or column wise. Being the similarity matrix symmetric row or column addition will yield the same result. The link weight can be considered as recommendation of one sentence by another and thus importance of a sentence is given by summation of link weights [16].

Degree of Centrality Method

Let us now consider degree of centrality method with a specified threshold proposed by Erkan et al. Here centrality degree" of any node is the number of edges incident on the vertex, with link weight greater than or equal to specified threshold. The idea behind this approach is to eliminate link weights which have too low values possibly noisy signals. If we choose a too high threshold the graph is not at all connected and becomes a set of islands.[16]

Lex Rank & Continuous Lex Rank methods

Each sentence in a network is considered as set of sentences. Each of these expressions, starts with arbitrary values assigned to each node in the graph, the computation iterates until convergence below a given threshold is achieved. After running the algorithm, a score is associated with each vertex, which represents the “importance” or “power” of that vertex within the graph. Each node is initially given a score of 1 / N where N is the number of sentences in each document. The lexical scores given are normalized by dividing each sentences weight with the maximum sentence weight, so that the top sentence score will be 1[15, 17].

Discounting Method

Discounting technique envisages that once a sentence is selected by any one of the methods, immediately corresponding row and column values of the matrix are set to zero [17]. Thus the next sentence is selected from contributions made by the remaining (n-1) sentences only [15]. The idea behind discounting technique is that once the sentence is selected, the chance for repetition of information in the succeeding sentences is minimized.

There have been a number of researches and development budgets [1] devoted to automatic text summarization. The United States (e.g., DARPA), the European Community and Pacific Rim countries have identified text summarization as a critical research area, and are investing init. Text summarization is also increasingly being exploit in the commercial sector, in tele communication industry (e.g., BT's Pro Sum), in filters for web based information retrieval (e.g. Inxight's summarizer used in Alta Vista Discovery), and in word processing tools (e.g.., Microsoft's AutoSummarize). In addition to the traditional focus of automatic abstracting (of scientific and technical text) to support information retrieval, researchers are investigating the application of this technology to a variety of new and challenging problems, including multi lingual summarization, multi media news broadcasts, and providing physicians with summaries of on-line medical literature related to patient's medical record. As the information overload problem has grown, and people become increasingly mobile and information-hungry, new applications for text summarization can be expected.

Performance Measures for Information Extraction

The results of information extraction module are evaluated against three most popular performance measures: recall, precision and f-measure.

Recall is the fraction of expert summary which is present in the summary generated by the system and is given as:

$$\text{Recall} = \frac{\text{No of Sentences (System Summary} \cap \text{Expert Summary)}}{\text{Total no of Sentences in Expert Summary}}$$

Precision is the fraction of the sentences extracted in the system summary that are present in the expert summary and is given as

$$\text{Precision} = \frac{\text{No of Sentences (System Summary} \cap \text{Expert Summary)}}{\text{Total no of Sentences in System Summary}}$$

F-measure is the weighted harmonic mean of precision and recall. The general formula for F-measure is given as

$$\text{F-Measure} = \frac{(1+\beta^2)*\text{Precision}*\text{Recall}}{\beta^2*\text{Precision}+\text{Recall}}$$

We have used the traditional (β =1) F-measure for our evaluation. This is also known as the F1 measure, because recall and precision are evenly weighted.

References

1) Mani, I. “Automatic Summarization”. John Benjamin’s publishing Co., limited edition 2001.
2) Udo Hahn, Inderjeet Mani. "The Challenges of Automatic Summarization". Computer, vol. 33, no. 11, Nov., 2000, pp 29-36.
3) Barzilay, R., McKeown, K. R., and Elhadad, M. “Information fusion in the context of multi-document summarization”. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics (College Park, Maryland, June 20 -26, 1999). Annual Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, 1999, pp 550-557.
4) Luhn, H. P. “The Automatic Creation of Literature Abstracts”. IBM Journal of Research and Development 2(2), 1969
5) Goldstein, J., Mittal, V., Carbonell, J., and Kantrowitz, M. “Multi-document summarization by sentence extraction”. In NAACL-ANLP 2000 Workshop on Automatic Summarization -Volume 4 (Seattle, Washington, April 30 -30, 2000). ANLP/NAACL Workshops. Association for Computational Linguistics, Morristown, NJ, 2000, pp 40-48.

6) Knight, K. and Marcu, D. “Summarization beyond sentence extraction: a probabilistic approach to sentence compression”. Artificial Intelligence 139, 1 (Jul. 2002), pp 91-107.

7) D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Celebi, E. Drabek, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel and Z. Zhang, "MEAD -a platform for multidocument multilingual text summarization," in LREC 2004, 2004.

8) Yeh, J., Ke, H., Yang, W., and Meng, I. “Text summarization using a trainable summarizer and latent semantic analysis”. Information Processing and Management. 41, 1 (Jan. 2005), pp 75-95.

9) Mihalcea, R. “Graph-based ranking algorithms for sentence extraction, applied to text summarization”. In Proceedings of the ACL 2004 on interactive Poster and Demonstration Sessions (Barcelona, Spain, July 21 -26, 2004). Annual Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, 2004, pp 20.

10) G Erkan, DR Radev, “LexRank: Graph-based Lexical Centrality as Salience in Text Summarization”, Journal of Artificial Intelligence Research, 2004.

11) Yang-Wendy Wang, “Sentence Ordering for Multi Document Summarization in Response to Multiple Queries”. A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the School of Computing Science, Simon Fraser University, 2006.

12) Nenkova and L. Vanderwende, "The impact of frequency on summarization," Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101, 2005.

13) Ramanathan, K., Sankarasubramaniam, Y., Mathur, N., Gupta, A. “Document Summarization using Wikipedia” (2009). In the proceedings of the First International Conference on Intelligent Human Computer Interaction (IHCI 2009), Springer, Jan 20-23 2009, pp 254-260.

14) Radev, D. R., Blair-Goldensohn, S., & Zhang, Z, “Experiments in Single and Multi-document summarisation using MEAD”, In Proceedings of the Document Understanding Conferences- 2001, 2001.

15) Gunes Erkan and Dragomir R. Radev (2004), “LexRank: Graphbased Lexical Centrality as Salience in Text Summarization”, Journal of Artificial Intelligence Research, pp. 457-479.]

16) Shanmugasundaram Hariharan and Rengaramanujam Srinivasan, “Studies on Graph Based Approaches for Single and Multi Document Summarizations”, International Journal of Computer Theory and Engineering, Vol.1, Issue No. 5, December 2009.

17) Shanmugasundaram Hariharan, Rengaramanujam Srinivasan Enhancements to Graph based methods for Multi Document Summarization Journal of Applied Computer Science, no. 6 (3) /2009.

7/18/2011