International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 1 144 – 149

An Effective Sentence Ordering Approach For Multi-Document Summarization Using Text Entailment

P Sukumar
Department of Computer Science
Sri Venkateswara College of Engineering
Chennai, India
spsukumaran@gmail.com

K S Gayathri
Department of Computer Science
Sri Venkateswara College of Engineering
Chennai, India
gayasuku@svce.ac.in

***Abstract*—** With the rapid development of modern technology electronically available textual information has increased to a considerable amount. Summarization of textual information manually from unstructured text sources creates overhead to the user, therefore a systematic approach is required. Summarization is an approach that focuses on providing the user with a condensed version of the original text but in real time applications extended document summarization is required for summarizing the text from multiple documents. The main focus of multi-document summarization is sentence ordering and ranking that arranges the collected sentences from multiple document in order to generate a well-organized summary. The improper order of extracted sentences significantly degrades readability and understandability of the summary. The existing system does multi document summarization by combining several preference measures such as chronology, probabilistic, precedence, succession, topical closeness experts to calculate the preference value between sentences. These approach to sentence ordering and ranking does not address context based similarity measure between sentences which is very essential for effective summarization. The proposed system addresses this issues through textual entailment expert system. This approach builds an entailment model which incorporates the cause and effect between sentences in the documents using the symmetric measure such as cosine similarity and non-symmetric measures such as unigram match, bigram match, longest common sub-sequence, skip gram match, stemming. The proposed system is efficient in providing user with a contextual summary which significantly improves the readability and understandability of the final coherent summary.

Keywords*-text summarization; preference experts; sentence ranking; sentence ordering; text entailment.*

*****

I. INTRODUCTION

Electronic document information in the web growing rapidly where time is a critical resource. The large volume of information is available to the user on a specific topic. It is not possible for a user to go through all the information and also nobody interested in reading all the contents and getting necessary information. It creates overhead to the user and also time consuming process. To help the user to obtain the necessary information in shortest time, a system should be designed that automatically process the information and converts it into a user efficient format. One solution is to provide the user with a condensed version of the original text.

Document summarization is a systematic activity aimed at extraction of required information from multiple texts written about the same topic. Single document and Multi-document summarization are the two basic types of document summarization. Single document summarization generates a single coherent summary from a single document. Multi-document summarization generates a single coherent summary from a given set of document that describes a particular event. In single document summarization, the extracted information will be in the same order as in the original document. In contrast multi-document summarization poses a number of new challenges such as identifying repetitions across various input documents, determining which information is to be included in the summary, organizing the selected information to create output summary.

Ordering is the process of putting the extracted sentences in proper order that in turn significantly improves the readability of documents. It is essential to pay attention to sentence ordering in case of multi-document summarization. Sentence position in the original document, which yields a good clue to sentence arrangement for single-document summarization, is not enough for multi-document summarization because inter-document order must be considered. In this paper, we focus on the sentence ordering problem in multi-document summarization.

The task of constructing a coherent summary from an extracted sentence has several unique properties that make it challenging to generate it. Source documents for a summary may have been written by different persons, have different texting styles, or written on different time periods, and based on different background knowledge. For example, a multi-document summarization system is presented with multiple texts that discuss about a particular news event. Those news texts are selected from different newspapers. Although the articles themselves are related and discuss a particular event, those articles are written by different persons at different times. Therefore, the collection of texts that the multi-document summarization system receives is not always coherent with regard to their authorship. Therefore we cannot assure set of extracted sentences from different unique documents to be coherent on their own.

Ordering extracted sentences from set of documents into a coherent summary is a non-trivial task. Rhetorical relations such as cause-effect relation and elaboration relation exist between sentences in a coherent text. If it is possible to determine the cause-effect relation directly that exists among a given set of sentences, then we can use those relations to infer

a coherent ordering of the set of sentences. For example, if a sentence *A* is the effect of the cause mentioned in a sentence *B*, then we might want to order the sentence *A* after sentence B in a summary that contains both sentences *A* and *B*. Unfortunately, the problem of automatically detecting the cause-effect relation of an arbitrary text is a very complex task.

We propose a novel approach for sentence ordering that identify the cause-effect relation in text by using symmetric and non-symmetric measures. Hence, the proposed system find out the contextual relationship between the sentences in the summary which in turn used to find the logical inferences between the sentences. The symmetric similarity measures is presented by standard cosine similarity measures and the non-symmetric is presented by finding the casual relation between the sentences in the summary.

II. RELATED WORK

Sentence ordering is a major issue in multi-document summarization for creating coherent summary. Multi-document summarization is useful in various applications such as Information access, automated ad placement, social media monitoring, and sentiment analysis toolkit to produce summary about particular event or topic. A number of methods related to sentence ordering have been developed recently. An effective similarity measure should be able to determine whether the sentences are semantically equivalent or not, taking into account the variability of natural language expression. That is, the correct similarity judgment should be made even if the sentences do not share similar surface form.

There are several similarity measures are used to find the relationship between the sentences in the summary. Achananuparp et al evaluate fourteen text similarity measures such as word overlap, Novelty Detection and Identity Measure, Linguistic Measures, TF-IDF Measures etc., which have been used to calculate similarity score between sentences in many text applications in order to know whether the sentences are semantically related or not.

A bottom up approach by Bollegala et al to sentence ordering for multi–document summarization [3] using supervised learning approach that concatenate four measures such as chronology, topical closeness, precedence, succession experts considered to arrange sentences using integrated strategy. In [2], suggested a preference learning approach to sentence ordering for multi–document summarization. This work used preference experts such as chronology, probabilistic, topical closeness, precedence, and succession to find similarity between sentences in the summary. Greedy algorithm is used to find the total ordering among the sentences.

A reinforcement approach to tightly integrate ranking and clustering of sentences by exploring term rank distributions over the clusters by Cai et al was proposed [4]. Based on initial '*k*' clusters, ranking is applied separately, which serves as a good measure for each cluster. Then, a mixture model is used to decompose each sentence into a *k* – dimensional vector, where each dimension is a component coefficient with respect to a cluster, which is measured by rank distribution. Then sentences are reassigned to the nearest cluster under the new measure space to improve clustering.

In order to find the similarity between two sentences more accurately semantic relationship should be considered [6]. In this work sentences would be divided into segments by some grammar rules, and each segment might be divided into several shorter segments. When calculating the semantic similarity between two sentences, the grammatical and semantic structure of the sentences would be analyzed, and the reasonable grammatical orders for segments in the two sentences would be chosen.

In the work [10], Peng et al proposed method for sentence ordering by using support vector machine. This work classify the sentences in the documents according to the source documents and adjust the sentence order based on the directional relativity of adjacent sentences, and the sequence of each group is found. Then, the sequences of different groups are connected to create the final order of the summary.

Interrelationship between texts units, including the correlation between units are calculated by hierarchical topic tree. The rhetorical relationship and temporal relationships were represented at different levels of granularity [13]. A series of algorithms including building Multi–document Rhetorical Structure (MRS), multi–document information fusion based MRS and summarization generation are also proposed.

All the above existing approach to sentence ordering in multi-document summarization does not addressed the semantic relationship between sentences in the summary which is very important to generate meaningful summary. The proposed system mainly focuses on finding the semantic relationship between sentences by building text entailment system model.

III. PROPOSED SYSTEM

A. Problem Staement

In multi-document summarization, the task of arranging the extracted sentences from multiple documents is very difficult. Because the sentences are written by various author in different period of time. The proposed system provides efficient methods to create coherent summary.

B. Senetence Ordering in Multi-Document Summarization

The main challenge of natural language processing such as Information Retrieval (IR), Information Extraction (TE), and Question Answering (QA) is provide computer system with the linguistic knowledge in order to perform language oriented task.

Text entailment is the process of finding the directional relationship between pair of sentences. The meaning of one sentence should be derived from another sentence, then we can say one sentence entails another sentence. For example the meaning of sentence *j* can be derived from meaning of sentence *i*, we can say that the sentence i entails sentence *j*.

In multi-document summarization sentence ordering is one of the major issue to deal with in natural language processing. For sentence ordering we proposed an approach that incorporates contextual relationship between sentences in the summary.

The figure 1 shows the overall framework for sentence ordering to generate ordered summary from an unordered sentences. Each ordering strategies are independent of each other which we call it as experts. The set of unordered sentences are given to each of the experts and it will return preference value of one sentence over another sentence as values in the range 0 to 1. [3] The preference values are calculated using the preference function as follows,

$$\mathrm{PREF}_e(u,v,Q) \in [0,1] \quad (1)$$

In Equation (1), *u, v* are two sentences to be ordered, *Q* is the set of sentences which has been ordered so far.

The preference of ordering *u, v* will be returned by the expert. If the expert prefers $u - v$ then it returns a value greater than 0.5. In the extreme case where the expert is absolutely sure of preferring $u - v$ it will return the value 1. On the other hand, if the expert prefers $v - u$ it will return a value less than 0.5. In the extreme case where the expert is absolutely sure of preferring $v - u$ it will return 0. When the expert is undecided of its preference between *u* and *v* it will return 0.5.

The proposed system includes five preference experts such as chronology, topical closeness, probabilistic, precedence, succession, text entailment experts. Chronology expert arranges the sentences according to the dates on which the documents were published. Publication timestamps are used to decide the chronological order among sentences extracted from different documents. In topical closeness expert, sentences conveying information to a particular topic tend to appear together in the summary. Therefore a coherent summary can be created by grouping sentences which are topically related.

The information stated in the document from where the sentence was extracted is be considered to judge the order. If the sentences preceding the extracted sentence in the original document match well with the so far ordered summary, it is suitable to order the sentence next in the summary. Precedence and succession relations are used to find the ordering among the sentences in a better way. The probabilistic expert using past history to order the sentences.

The text entailment expert system is used to find the logical relationship between two sentences in the summary. The proposed entailment system uses various symmetric and non-symmetric measures to find the entailment among sentences.

The linear weighted sum of these individual preference functions is taken as the total preference by the set of experts as follows,

$$\mathrm{PREF}_{\mathrm{total}}(u,v,Q) = \sum_{e \in E} W_e \mathrm{PREF}_e(u,v,Q) \quad (2)$$

In Equation (2), *E* is the set of experts and w_e is the weight associated with expert $e \in E$. These weights are normalized such that the sum of them equals to 1.

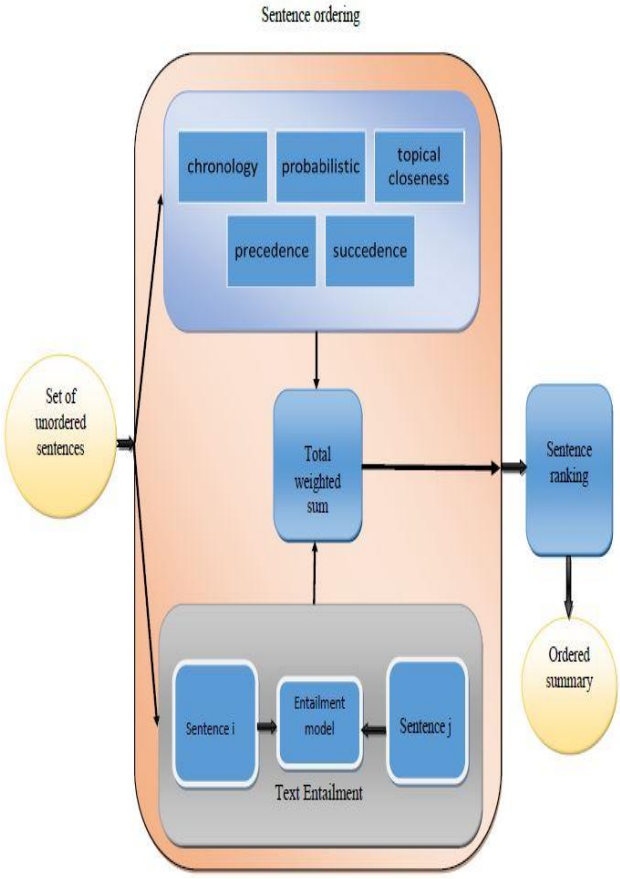

Figure 1: Overall Framework for Sentence Ordering

C. *Preprocessing*

The very basic step is splitting the sentences of the summary into smaller number of units called tokens. It may contain words, numbers, punctuation, etc. From list of tokens remove the terms which does not contribute to the meaning of the sentence which avoids the processing overhead by processing unwanted words.

D. *Similarity Measures*

There are two similarity measures such as symmetric and non-symmetric are considered here for finding entailment between sentences in the summary. The machine learning algorithm is trained using these two features. The classifier return either "yes" or "no" based on the entailment decision.

i) *Symmetric Measures*

Cosine similarity measure is used to find the similarity between two n-dimensional vector obtained by finding the cosine angle. It is used to find the similarity between sentences in the summary in case of natural language processing. Given two vectors of attributes, S_j and S_k, the cosine similarity θ is calculated using the dot product and magnitude as,

$$\mathrm{sim}(s_j, s_k) = \frac{\vec{s_j} \cdot \vec{s_k}}{\left|\vec{s_j}\right| \left|\vec{s_k}\right|} \quad (3)$$

In Equation (3), S_j and S_k represents sentences in the document collection or summary.

ii) Non-symmetric measure

The casual relation between sentences in the summary is considered here. There are few lexical similarity measures are taken into account to find out the casual relationship between sentences (cause-effect relation). As per Hobbs's casual relation [14] if a segment stating a cause occurs before a segment stating an effect. For example sentence *B* is the necessary cause of sentence *A* in a summary that contains both the sentences, the sentence *A* should be ordered before the sentence *B* in the summary. In the below section we see some the lexical feature used for finding the entailment pairs in the summary.

1) Lexical unigram match

In this method the presence of various unigrams in the sentence *i* is checked against the sentence *j* with each text hypothesis pair (sentence *i* - sentence *j*). WordNet synsets are used for identifying unigrams match.
For example, consider a pair of sentences in a particular summary.

Sentence *i* - The formula of richter scale was designed by US seismologist Richter in 1935.
Sentence *j* - The Richter scale was created by US seismologist Richter in 1935.
Here, Common unigrams are Richter, scale, US, seismologist, and 1935

If *n1* = common unigram between sentence *i* and sentence j and *n2* = number of unigrams in sentence *j*, then

$$Lex\_unigram\_match = n1/n2. \quad (4)$$

If the value of Lex_unigram_match is 0.75 or more, i.e., 75% or more unigrams in the sentence *j* is matches either directly or through WordNet synonyms of sentence *i*, then the pair of sentence considered as an entailment. The preference function will return the value 1 if entailment is true, otherwise it will return 0.

2) *Lexical bigram match*

Each bigram in the sentence *j* is searched for their presence in the corresponding sentence *i* part. The Lex_bigram_match is calculated as follows, i.e.,

$$Lex\_bigram\_match = n1/n2. \quad (5)$$

Where, *n1* is the total number of bigram match between sentence *i* and sentence *j* pair and *n2* is total number of bigrams in the sentence *j*.

If the value of Lex_bigram_match is 0.50 or more, i.e., 50% or more bigrams in the sentence *j* is matches with bigrams in the sentence *i*, then the sentence pair is considered as an entailment pair. The preference function will return the value 1 if entailment is true, otherwise it will return 0.

3) *Lexical longest common subsequence*

The longest common subsequence of sentence *i* – sentence *j* pair is the longest sequence of words that is common to both sentences. The Lex_LCS_match is calculated as follows, i.e.,

Lex_LCS_match= LCS (sentence i, sentence j)/length of unigrams in the sentence j. (6)

If the value of Lex_LCS_match is 0.80 or more, i.e., the length of the common words in pair of sentence is greater than the length of the sentence *j*, then the sentence pair is considered as an entailment pair. The preference function will return the value 1 if entailment is true, otherwise it will return 0.

4) *Lexical skip gram match*

A skip gram is any combination of words in the order as they appear in a sentence but allowing gap between word occurrences. In the proposed work 1-skip bigram is considered where 1-skip bigram allowing one word gap between words in a sentence as they appear.

$$Lex\_1\_skip\_bigram\_match = n1/n2. \quad (7)$$

Where, *n1* is the 1_skip_bigram_match between sentence *i* and sentence *j* and *n2* is the total number of unigrams in the sentence j.

If the value of Lex_1_skip_bigram_match is 0.50 or more, i.e., 1-skip bigram match between sentence *i* and sentence *j* is greater than the length of the sentence *j*, then the sentence pair is considered as an entailment pair. The preference function will return the value 1 if entailment is true, otherwise it will return 0.

5) *Lexical stemming match*

Getting the stem of each word in the sentence *j* by reducing terms to their root forms. For example, the plural forms of a noun such as 'drugs' are stemmed into 'drug', and mostly ending with 'ing', 'es' , 's' , 'ed' are removed from verbs. Each word in the sentence pair is stemmed using the stemming function provided along with the WordNet 2.0.

If *n1* = Number of common stemmed unigrams between sentence *i* and sentence *j* and *n2* = Number of stemmed unigrams in the sentence *j*, then Lex_stem_match is calculated as follows i.e.,

$$Lex\_stem\_match = n1/n2. \quad (8)$$

If the value of Lex_stem_match is 0.7 or more, i.e., 70% or more stemmed unigrams in the sentence *j* match in the stemmed sentence *i*, then the sentence pair is considered as an entailment pair. The preference function will return the value 1 if entailment is true, otherwise it will return 0.

E. Ordering Algorithm

Using the five preference functions in the previous part we compute the total preference function using the equation (2). The problem of finding optimal ordering for a given total preference function is done by sentence ordering algorithm.

Given an unordered sentences *X* extracted from a set of documents, and total preference function, $PREF_{total}(u,v,Q)$ computes a total ordering function among the extracted sentences.[3] The sentence ordering algorithm is given below,

Algorithm 1: Sentence Ordering Algorithm

Input: A set x of the extracted (unordered) sentences and a total preference function

Output: Ranking score p of each sentence t € x

1. V=x
2. Q= ∅
3. **for each** $v \in V$ do
4. $\Pi(v)=\Sigma_{u\in V} PREF_{total}(v,u,Q)-\Sigma_{u\in V} PREF_{total}(u,v,Q)$
5. **end for**
6. **while** $V \neq \varnothing$ do
7. $t=\text{argmax}_{u\in V}\prod(u)$
8. $\rho(t)=|V|$
9. V=V-{t}
10. Q=Q+{t}
11. **for each** $v \in V$ do
12. $\Pi(v)=\Pi(v)+PREF_{total}(t,v,Q)-PREF_{total}(v,t,Q)$
13. end for
14. end while
15. return p

The line 1 assign the set of unordered sentences to the set V. In the line 2 we assign the initial count of sentences in the ordered summary is equal to null. From the line number 3 to 10 the algorithm does, for each sentence t the function ρ(t) will be calculated and it returns the rank value. The sentence with highest rank will be choose first and added into the ordered summary. From line 11 to 15 we have to re-calculate preference values of all sentences. The preference values of sentences are calculated by comparing sentence which resides already in the ordered summary against sentences further wants to be ordered and the sentence with next high rank value will be added further into the summary. This above is process repeated until there is no more sentences to order.

IV. EXPERIMENTAL RESULTS

Text Analysis Conference (TAC) released data set for evaluation in the year 2008. The input documents for the proposed method are taken from TAC 2008 AQUAINT–2 collection of newswire articles. AQUAINT–2 collection of news articles span from October 2004 to March 2006 with 48 topics and each topic consists of 10 documents and its summary.

To generate the training data for proposed system, human annotators are asked to arrange the extracted sentences. Here, two human annotators worked independently and arranged sentences extracted for each topic. They were provided with the source documents before ordering sentences in order to gain the background knowledge on the particular topic. From the manual ordering process we obtained 48*2=96 set of ordered extracts.

To produce sentence orderings we select the set of extracted sentences for one topic as test data and remaining 47 as training data and repeat this process 48 times by selecting a different set at each round. To evaluate the performance of the proposed sentence ordering approach, we compare the result of the proposed method with the existing method. Automatic system generated summary is compared against the human annotators generated reference summary. Precision is the parameter used for the evaluation, can be calculated as follows,

$$P=m/N-n+1 \quad (9)$$

In (9), P is the precision value, m is the number of continuous sentences appear in both reference and system generated summary, n is length of continuous sentences and N is the number of sentences in the reference ordering.

Three methods are used to order the extracts. Random ordering, Chronology ordering, Learned ordering. Chronology, probabilistic, succession, precedence, topical closeness experts are combined in case of learned ordering. For example when we consider 3 continuous sentences for calculation, the precision value for random ordering is 0, for chronology the precision value is 0.49 and for learned ordering the value is 0.55. Further, we add text entailment expert in the learned ordering to generate effective summary with higher precision value.

V. CONCLUSION AND FUTUREWORK

The proposed entailment model provides a systematic approach for sentences ordering and ranking for multiple documents. A graph model is used for sentence ranking where each nodes represents sentences and edges represents preference value between sentences. The preference value are calculated using chronological, probabilistic, topical closeness, precedence, succession and text entailment experts. Text entailment model addresses the contextual relationships between sentences through cause and effect relation approach using symmetric and non-symmetric measures. This method provides high accuracy compared to statistical methods by providing efficient contextual summary which significantly improves readability and understandability. In future, the syntactic non-symmetric measures will also be taken into account to produce more effective summary.

REFERENCES

[1] P. Achananuparp, X. Hu and X. Shen, "The evaluation of sentence similarity measures", 10th International Conference on Data Warehousing and Knowledge Discovery 2008.

[2] D. Bollegala, N. Okazaki and M. Ishizuka, "A bottom-up approach to sentence ordering for multi-document summarization", Information Processing and Management, 2010, Vol.46, No.1, pp. 89–109.

[3] D. Bollegala, N. Okazaki and M. Ishizuka, "A preference learning approach to sentence ordering for multi-document summarization", Information Science, 2012, pp. 78-95.

[4] X. Cai and W. Li, "Ranking Through Clustering: An Integrated Approach to Multi-Document Summarization", 2013, Vol.21, No.7, pp. 1424-1433.

[5] J J. Castillo, "An approach to Recognizing Textual Entailment and TE Search Task using SVM", Natural Language Processing, 2010, Vol. 44, pp. 139-145.

[6] Y. Liu and Y. Liang, "A sentence semantic similarity calculating method based on segmented semantic comparison", Journal of Theoretical and Applied Information Technology, 2013, Vol. 48, No. 1, pp. 231-235.

[7] Miguel Angel Rios Gaona, Alexander Gelbukh, and Sivaji Bandyopadhyay, "Recognizing Textual Entailment Using a Machine Learning Approach", 2011.

[8] M. Lapata, "Probabilistic text structuring: experiments with sentence ordering", in: Proceedings of the Annual Meeting of ACL, 2003, pp. 545–52.

[9] P. Partha, B. Sivaji and G. Alexander, "Textual entailment using lexical and Syntactic similarity", International Journal of Artificial Intelligence & Applications, 2011, Vol. 2, No. 1, pp. 43-58.

[10] G. Peng, Y. He, Y. Tian, and W. Wen, "Analysis of Sentence Ordering Based on Support Vector Machine", Pacific-Asia Conference on Knowledge Engineering and Software Engineering, 2009.

[11] G. Peng, Y. He, W. Zhang, N. Xiong and Y. Tian, "A Study for Sentence Ordering Based on Grey Model", IEEE Asia-Pacific Services Computing Conference, 2010.

[12] J K. Yogan and N. Salim, "Automatic Multi Document Summarization Approaches", Journal of Computer Science, 2012, Vol.8, No.1, pp. 133–140.

[13] Yong-Dong Xu, Xiao-Dong Zhang, Guang-Ri Quan and Ya-Dong Wang, "MRS for multi-document summarization by sentence extraction", 2013.

[14] J. R. Hobbs, "Ontological promiscuity", Proceedings of the 23rd annual meeting on Association for Computational Linguistics, 1985.