Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. Engg. 40 (2011) 5372-5375

ISSN: 2229-712X

Semantic summary generation from multiple documents using feature specific sentence ranking strategy

A.Kogilavani and P.Balasubramanie

Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Tamilnadu -638052.

ARTICLE INFO

Article history:

Received: 4 October 2011;

Received in revised form:

22 October 2011;

Accepted: 3 November 2011;

Keywords

Dependency Parsing,

Feature Specific Sentence Ranking Strategy,

Semantic Similarity Matrix.

ABSTRACT

This paper proposes an approach of adapting the vector space model with dependency parse relations to generate semantic summary from multiple documents. Traditional vector space models with tf-idf weighting was not able to completely capture the content similarity because it treats the words within a document are independent of each other. In the proposed system the dependency parse of the document has been used to modify the tf-idf weight of words by incorporating the dependency between each pair of words. To select relevant sentences, different combinations of features are applied through sentence ranking strategy. The experiment result shows that consistent improvement of proposed system over traditional approaches.

Introduction

Electronic document information is exponentially growing and where time is a critical resource in this epoch, it has become practically impossible for any user to read large numbers of individual documents. It is therefore important to discover methods of allowing users to extract the main idea from collection of documents.

Automatic document summarization of multiple documents would thus be immensely useful to fulfill such information seeking goals by providing an approach for the user to quickly view highlights or relevant portions of documents. Multi-document summarization is the process of generating a summary by reducing documents in size while retaining the main characteristics of the original documents. Since one of the problems of data overload is caused by the fact that many documents share the same or similar topics, automatic multi-document summarization has attracted much attention in recent years.

The issues to be considered for multi-document summarization are as follows: First, simple word-matching measure is not able to completely capture the content similarity because news articles consist of different words to describe the same events.

Traditional vector space model assumes a bag-of-words model of the document where the words within a document are independent of each other. Therefore, effort has to be taken to find the dependency between the words which is used to select semantically important sentences from the document collection. Second, generating well organized fluent summary by selecting more relevant information from multiple documents. This can be done with the help of feature specific sentence ranking strategy.

Related Work

Summary is generated from multiple documents by constructing statistical vector space model and then modifying it using the concept of action words to form semantic vector space model as mentioned by Om, Akhil, Girraj, Amit (2008).

Action words are identified by consulting with knowledgebase consisting of seed word list which is generated manually. Latent Semantic Indexing is utilized to generate extractive summary from multiple documents in (Kiril, 2008). In this approach, no sophisticated syntactic, semantic analysis or natural language generation was involved.

In (Dingding, Tao, Shenghuo, Chris, 2008) sentence-sentence similarities are calculated using semantic analysis and from this similarity matrix is constructed.

Symmetric matrix factorization is used to group sentences into clusters. Finally most informative sentences are selected from each group to form the summary.

The approach proposed as in June, Hsin (2008) uses words and event words to deal with multi-document summarization. These words indicate the important concepts and relationships in a document or among a set of documents, and can be used to select salient sentences.

A summarizer produced by Harris, Oussalah (2008) which explicitly makes use of the semantic relatedness of document sentences using Word Net taxonomy. Yihong, Xin (2001) discusses about two methods that are used to create generic text summaries by ranking and extracting sentences from multiple documents.

This paper is organized as follows. Section III presents an overview of the proposed system. Section IV discusses about evaluation criteria and experimental results. Section V presents the conclusion.

Preprocessing

Tokenization is the very basic ability of splitting text in the documents into meaningful units like words, punctuation, etc. Split the documents into sentences and then into words.

From words list remove frequently occurring insignificant words called stop words because they do not contribute to the meaning of the sentence. Get the stem of each word by applying enhanced Porter Stemmer algorithm.

Tele: +9194861-53223

E-mail addresses: vani_sowbar@yahoo.co.in

Proposed system

Figure 1 illustrates an overview of the proposed approach for semantic summary generation.

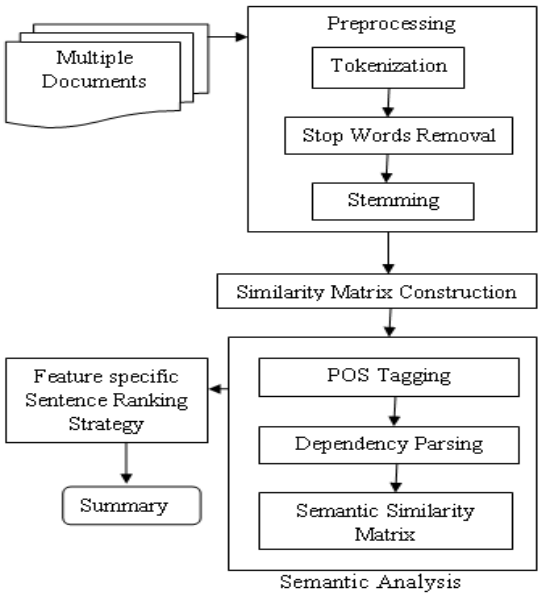

Figure 1. Proposed System

Similarity Matrix Construction

Let *D* be a collection of documents with common topics, *k* be the total number of documents in *D*, *N* be the number of all sentences in document collection, *m* be the number of words in each sentence, d_i be the i^{th} document in *D*, $S_{i,k}$ be the i^{th} sentence in document d_k, w be a word. The Term Frequency(*TF*) of each word is calculated by

$$TF(w_i) = \frac{n_j}{\sum_k n_k} \quad (1)$$

where n_j is the number of occurrences of the term *j* in the document and the denominator is the number of occurrences of all terms in the document collection. Inverse Document Frequency is calculated as,

$$IDF(w_i) = \log \frac{|D|}{d_i} \quad (2)$$

where d_i is number of documents that contain term *i* and |*D*| is total number of documents in the collection. For example, the similarity matrix constructed for the sentence "June 1994 : Airbus begins engineering development of the plane, then known as the A3XX." is given in Table 1.

Semantic Analysis

Semantic analysis algorithm

- Get the similarity matrix of document collection D.
- Assign Part-of-Speech (PoS) to each word in the document to get tagged documents.
- Apply tagged documents to parser to find the dependencies between words in a sentence.
- Identify verbs in each sentence of the document and objects (nouns/adjectives) that are affected by it.
- Find contextual object. If more than object is there for the same verb then maximum weight amongst all the objects corresponding to the given verb is selected as contextual object.
- Add the contextual object weight with its similarity matrix weight to form semantic similarity matrix.

PoS tagging

In corpus linguistics, part-of-speech tagging is the process of marking up the words in a corpus as corresponding to a particular part of speech, based on both its definition, as well as its context relationship with adjacent and related words in a sentence. In order to assign parts of speech to each word the proposed system utilizes Stanford Log-Linear Part-of-speech Tagging as in Marneffe, Maccartney (2008) produces tagged documents. Then the tagged documents are passed through Stanford parser to extract grammatical relationships in a sentence and the output is represented using Stanford typed dependencies. For the above sample sentence the tagged output are:

June/NNP 1994/CD:/: Airbus/NNP begins/VBZ engineering/NN development/NN of/IN the/DT plane/ NN ,/, then/RB known/VBN as/IN the/DT A3XX/NN

Dependency parsing

A dependency parse represents dependencies between individual words. A typed dependency parse additionally labels dependencies with grammatical relations such as subject and indirect object. Each word in the sentence is the dependent of one other word. For the above sample sentence the typed dependencies are represented as:

June(1) 1994(2) :(3) Airbus(4) begins(5) engineering(6) development(7) of(8) the(9) plane(10) ,(11) then(12) known(13) as(14) the(15) A3XX(16)

Stanford dependencies generated for each of the above parsed sentences carry word-position numbers along with their arguments. Typed-dependencies of the above sentence is presented in Table 2.

Semantic similarity matrix construction

For the above sentence, first verb-object pair is identified and then contextual object is identified if more than one object is there for the same verb. Table 3 and Table 4 presents object-verb pair and contextual object information for the sample sentence. . The object with maximum weight is added with the original object weight to modify the weight of the corresponding verb which is represented in Table 5.

Summary generation by sentence ranking strategy

To capture the relevant sentences from multiple documents, the proposed work combines six features from Kogilavani, Balasubramanie (2010) with additional features like word similarity between sentence and topic, sentence frequency score and document frequency score.

Word feature

$$W\_F(s_{i,k}) = \sum Word\_Score(s_{i,k}).f(w_m, s_{i,k}) \quad (3)$$

$$Word\_Score(s_{i,k}) = \sum_{i=1}^{m} S\_(TF(w_i).IDF(w_i)) \quad (4)$$

Position feature

$$P\_F(s_{i,k}) = \frac{Position(s_{i,k})}{3} \quad (5)$$

Sentence length feature

$$L\_F(s_{i,k}) = \frac{N * length(s_{i,k})}{length(d_k)} \quad (6)$$

Sentence centrality feature

$$C\_F(s_{i,k}) = \frac{words(s_{i,k}) \cap words(others)}{words(s_{i,k}) \cup words(others)} \quad (7)$$

Sentence with proper noun feature

$$PN\_F(s_{i,k}) = \frac{PN\_Count(s_{i,k})}{Length(s_{i,k})} \quad (8)$$

Sentence with numerical data feature

$$ND\_F(s_{i,k}) = \frac{ND\_Count(s_{i,k})}{Length(s_{i,k})} \quad (9)$$

Word similarity between sentence and topic feature

Any sentence that contains words similar to the given topic is an important one. To identify the similarity between the term and the topic, the following eq.(10) is used.

$$WSim\_F(s_{i,k}) = \sum_{w_i \varepsilon T, w_j \varepsilon S} sim(w_i, w_j) \quad (10)$$

where $sim(w_i,w_j)$=1 if both word and the topic are same, 0 otherwise. Here T represents topic sentence.

Sentence frequency score feature

To determine sentence frequency score, the following eq.(11) is used. To calculate the importance of individual word in a sentence, eq.(12) is used.

$$SFS(s_{i,k}) = \sum_{i \in S} \frac{SFS(w_i)}{|S|} \quad (11)$$

$$SFS(w) = \frac{\{|s|: w \in s\}}{|N|} \quad (12)$$

Document frequency score feature

To determine document frequency score, the following eq.(13) is used. To calculate the importance of individual word in a sentence, eq.(14) is used.

$$DFS(w) = \frac{\{|d|: w \in d\}}{|D|} \quad (13)$$

$$DFS(s_{i,k}) = \sum_{i \in d} \frac{DFS(w_i)}{|d|} \quad (14)$$

Finally sentence score is calculated for each sentence based on the following eq.(15).

$$Sentence\_score(s_{i,k}) = W\_F(s_{i,k}) + P\_F(s_{i,k}) + L\_F(s_{i,k}) + C\_F(s_{i,k}) + PN\_F(s_{i,k}) + ND\_F(s_{i,k}) + WSim\_F(s_{i,k}) + SFS(s_{i,k}) + DFS(s_{i,k}) \quad (15)$$

Sentence score is calculated for all sentences in different feature combinations. High scored sentences are selected for summary and those sentences are arranged in decreasing order of score. Highest ranking sentences are selected and summary is generated by arranging the selected sentences in the order in which they appeared in original documents.

Experiments and Evaluation

The documents for summarization are taken from the AQUAINT-2 collection of newswire articles. The AQUAINT-2 collection comprises news articles spanning the time period of October 2004-March 2006. Articles are in English and 48 topics were there and each topic consists of 20 documents and divided into two sets of 10 documents each, such that Set B followed Set A in the temporal order. For this work, Set A documents are utilized to generate summary.

Evaluation Measure

Precision

Precision can be calculated based on machine generated summary and the human summary. Precision (P) is defined as

$$P = \frac{N_o}{N_m}$$

where N_o = Number of common terms in both human and machine summary, N_m= Number of terms in machine summary.

Recall

Recall (R) is defined as

$$R = \frac{N_o}{N_h}$$

where N_o = Number of common terms in both human and machine summary, N_h= Number of terms in human summary.

F_Meassure is weighted arithmetic mean of Precision and Recall. Figure 2 represents Precision, Recall and F_Measure values calculated by TF-IDF and S_(TF-IDF). The result shows that through S_(TF-IDF) semantics of sentence is utilized to generate summary.

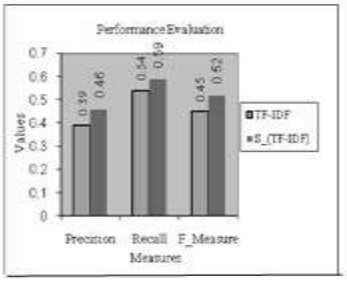

Figure 2. Comparison of TF-IDF and S_(TF-IDF)

ROUGE-1 measure

ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation.

$$ROUGE\_1 \ \ Score = \frac{X}{Y}$$

where *X* is count of number of unigrams that occur in machine and manual summary and *Y* is total number of unigrams. The following table 6 compares ROUGE-1 Score of proposed system against MEAD approach as in Radev, Jing, Sty,Tam (2004). The result shows that by utilizing S_(*TF-IDF)* and sentence specific features, the proposed system machine generated summary improves the accuracy of the summary.

Conclusion

The proposed system extracts sentences from multiple documents based on semantic analysis and relevant sentences are selected by applying different combinations of features. Relevancy is improved by employing S_(TF-IDF) measure. The summary generated using the proposed method is compared with human summary and its performance has been evaluated and the result shows that the summary generated by the proposed system is efficient compared with existing system.

References

[1] Om Vikas, Akhil K Meshram, Girraj Meena. Multiple Document Summarization Using Principal Component Analysis Incorporating Semantic Vector Space Model. Computational Linguistics and Chinese Language Processing. 2008; 13(2): 141-156.

[2] Kirill Kireyev. Using Latent Semantic Analysis for Extractive Summarization. Proceedings of Text Analysis Conference; USA; 2008.

[3] Dingding Wang , Tao Li , Shenghuo Zhu , Chris Ding. Multi-Document Summarization via Sentence-Level Semantic Analysis and Symmetric Matrix Factorization. Proceedings of ACM SIGIR conference on Research and development in information retrieval; New York, USA; 2008.

[4] June-Jei Kuo, Hsin-Hsi Chen. Multidocument Summary Generation: Using informative and event words. ACM Transactions on Asian Language Information Processin. 2008; 7(1).

[5] Harris, Oussalah. Automatic document summarizer. 7th IEEE International Conference on Cybernetic Intelligent Systems; London; 2008.
[6] Yihong Gong, Xin Liu. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. Proceedings of ACM SIGIR conference on Research and development in information retrieval; New York, USA; 2001.
[7] Marneffe, Maccartney, Manning. Stanford typed dependencies manual. 2008.
[8] Kogilavani, Balasubramanie. Clustering and Feature Specific Sentence Extraction Based Summarization of Multiple Documents. International Journal of Computer Science and Information Technology. 2010; 2(4): 99-111.
[9] Radev, Jing, Sty,Tam. Centroid-based summarization of multiple documents. Information Processing and Management.

Table 1. Similarity Matrix for sample sentence

Word	TF-IDF	Word	TF-IDF
June	0.0060165905110677	engineering	0.00300829525553389
1994	0.00430389750185574	development	0.00601659051106778
Airbus	0.00000000000000000	plane	0.00000000000000000
begins	0.00860779500371149	A3XX	0.00430389750185574

Table 2. Typed-dependencies for the sample sentence

Typed Dependencies	
num(June-1, 1994-2)	det(plane-10, the-9)
nsubj(begins-5, Airbus-4)	advmod(known-13, then-12)
dep(June-1, begins-5)	partmod(plane-10, known-13)
nn(development-7, engineering-6)	prep(known-13, as-14)
dobj(begins-5, development-7)	det(A3XX-16, the-15)
prep(development-7, of-8)	pobj(as-14, A3XX-16)

Table 3. Object-Verb List

Verb	Object
begins	Airbus June development

Table 4. Contextual Object

Verb	Weight
Begins	max(0.000000000000000,0.0060165905110677 8,0.0060165905110677 8)

Table 5. Semantic Similarity Matrix

Word	S_(TF-IDF)	Word	S_(TF-IDF)
June	0.0060165905110677	engineering	0.00300829525553389
1994	0.00430389750185574	development	0.00601659051106778
Airbus	0.00000000000000000	plane	0.00000000000000000
begins	0.01462438551477927	A3XX	0.00430389750185574

Table 6. Comparison of ROUGE-1 Score

Approach	ROUGE-1 Score
Existing System	0.455
Proposed System	0.598