A Hybrid Method for Query based Automatic Summarization System

R. V. V. Murali Krishna
G V P College of Engineering(A)
Visakhapatnam
AP, INDIA

S. Y. Pavan Kumar
G V P College of Engineering(A)
Visakhapatnam
AP, INDIA

Ch. Satyananda Reddy
Andhra University
Visakhapatnam
AP, INDIA

ABSTRACT

Automatic text summarization is one of the research goals of Natural Language Processing which relieves humans from studying each and every line in a text document to understand the underlying concepts in it. Automatic text summarization is aimed to create a brief outline of a given text covering the important points in the text. Automatic text summarization can be generic or query specific. This paper is focused on Query specific text summarization where a summary of the given text is constructed based on the given query. Query specific text summarization is based on the calculation of the relationship between sentences in the text document and the query given. Several statistical techniques and linguistic techniques have been developed to find the relationship between the given query and the sentences in the document. These methods when used alone could not give desired accuracy in the results. In this paper a sentence scoring method is defined based on existing sentence scoring methods. It attempts to combine the individual results of these methods to give a better assessment of the relationship between the sentences.

General Terms

Automatic text summarization ;Natural language processing; statistical techniques; linguistic techniques; stemming; n-gram; clustering

1. INTRODUCTION

Text summarization is the process of extracting key phrases/sentences from the given text and presenting them in a readable format to the user. The text presented to the user is called as summary. There are 2 types of summaries 1) indicative 2) informative. Indicative summaries do not carry the essence of the original text but give an idea about the information. Informative summaries carry the essence of the original text. There can be two types of informative summaries. They are 1) Extractive summary 2) abstractive summary. Extractive summaries carry the essence of the original text though the words/sentences present in the original text. Abstractive summaries carry only the essence in the original text but not its words/sentences.

1.1 Main tasks in extractive text summarization are

- *1.1.1 Keyword extraction*
- *1.1.2 Sentence scoring*
- *1.1.3 Sentence ordering*

1.2 Main tasks in abstractive text summarization are

- *1.2.1 Content selection*
- *1.2.2 Sentence compression*
- *1.2.3 Sentence fusion*
- *1.2.4 Sentence generation*

Various commercial tools are available for extractive text summarization. There are no commercial tools available for abstractive text summarization. Lot of research has been done on extractive summarization areas like keyword extraction, sentence scoring methods and sentence ordering.

Little research has been done in the areas like content selection, sentence compression, sentence fusion and sentence generation of abstractive text summarization. Depending on what the summarization program focuses to make the summary of the text, summaries can be generic or query relevant. Summary can be generated for an individual document or for a collection of documents. On the basis how summary is generated, Summary generation can be broadly divided as abstractive and extractive. In abstractive summary generation, the abstract of the document is generated. The summary so formed need not have exact sentences as present in the document. In extractive summary generation, important sentences are extracted from the document based on sentence similarity methods. The generated summary contains all such extracted sentences arranged in a meaningful order. This paper mainly focused on query specific extractive summarization of source document. Extractive techniques copy the information deemed most important by the system to the summary (for example, key clauses, sentences or paragraphs)

2. RELATED WORK

Gholamrezazadeh, Saeedeh et. Al have done comprehensive survey on text summarization systems. They have given a detailed understanding of various approaches used in text summarization systems. In statistical approaches the sentence selection is done based on statistics of the text like word frequency, position of the sentence, location of the word etc… [1].These methods are based on the idea that text surface cues are the most obvious indication of the text contents. There are several methods for determining the key sentences such as, The Title Method [2], The Location Method [3], The Aggregation Similarity Method [5], The Frequency Method [4], TF- Based Query Method [2] etc. Some examples of different systems which use these methods for generating a summary are IN XIGHT's [6] Summary Server is an application that creates extraction-based summaries offline. It uses statistical extraction techniques based on features such as sentence position, sentence length and keywords. University of southern California developed the SUMMARIST [6] system which produces summaries of web documents. It first identifies the main topics of the document using statistical techniques based on features such as position, and word counts.

Linguistic approaches are based on considering the semantic relationship between the sentences. This semantic relationship can be calculated by using several methods like Word Net [7], How Net, Google similarity distance [8] etc.

The following are the various statistical and linguistic techniques used in this paper for calculating similarity score between the two sentences.

2.1 Sentence scoring methods

Sentence scoring methods are very important in a document summarization. The efficiency of a summarization system mostly depends on the sentence scoring methods. The main task of the sentence scoring methods is to identify set of sentences which will carry important data in the given document. The scoring methods are also known as sentence similarity measures. The following are the various statistical and linguistic techniques used in this paper for calculating similarity score between the two sentences.

2.1.1 Statistical Techniques:

2.1.1.1 Word form similarity [9]

2.1.1.2 N-gram based similarity [12]

2.1.1.3 Word Order Similarity [10]

2.1.2 Linguistic Techniques:

2.1.2.1 Semantic similarity [10]

2.1.1.1 Word form similarity:

The word form similarity is mainly used to describe the form similarity between two sentences, is measured by the number of same words in two sentences. The sentences are preprocessed to filter the keywords from the overall words in the sentence. If S1 and S2 are two sentences, the word form similarity is calculated by using the below formula

Word form similarity (s1, s2) =

(2*Number of Same words (s1, s2)) / (Len (s1) + Len (s2)) *100

Example: The following example shows how the wordform similarity is calculated between a sample query and a sentence in a document

Query: Computer design contains memory chips. *Sentence*: Memory chips are the main parts of a computer.

Query is partitioned in to tokens: Computer |design|contains | memory | chips.

After applying stop list: Computer |design |memory |chips.
Applying Stemming: comput | design | memori | chip.

Sentence is partitioned in to tokens: Memory |chips| are |the|main |parts |of |a |computer.

After applying stop list: Memory| chips |computer.

Applying stemming algorithm: Memori| chip| comput.Calculation of Word Form Similarity for above two sentences:

Word form similarity (s1, s2) =

(2*Number of Same words (s1, s2)) / (Len (s1) + Len (s2)) *100

Similarity (S1, S2) = ((2*3)/ (4+3)) * 100
Similarity (S1, S2) =6/7=0.8571* 100 = 85.71%
Similarity (S1, S2) = 85.71%

Therefore we can say that the above two sentences are almost similar.

2.1.1.2. N-gram based similarity:

N-grams are fixed length consecutive series of "n" characters. An n-gram of size 1 is referred to as a "unigram", an n-gram of size 2 is a "bigram", an n-gram of size 3 is a "trigram", size 4 or more is simply called an "n-gram". The n-gram based similarity can be calculated after removing the stop words from the two sentences by using the formulae.

N-gram similarity (s1, s2) =

(2 * (no of common N-grams in s1 & s2)) / (Total no of N – grams in s1 & s2) * 100

Example: The following example shows how the N-grambased similarity is calculated between a sample query and a sentence in a document

Query: Computer design contains memory chips. *Sentence*: Memory chips are the main parts of a computer.

Query is partitioned in to tokens: Computer |design|contains | memory | chips.

After applying stop list: Computer |design |memory |chips.

Splitting the query in to bigrams:Computer: co |om |mp |pu |ut |te |er Design: de|es |si |ig |gn
Memory: me |em| mo| or| ry
Chips: ch |hi| ip |ps

Sentence is partitioned in to tokens: Memory |chips| are |the|main |parts |of |a |computer.*After applying stop list*: Memory| chips |computer.

Splitting the sentence in to bigrams:

Memory: me |em| mo| or| ry
Chips: ch |hi| ip |ps
Computer: co |om |mp |pu |ut |te |er

Calculation of bi-gram based Similarity for above two sentences as:

N-gram similarity (s1, s2) =
(2 * (no of common b_i - grams in s1 & s2))/(Total no of b_i – grams in s1 & s2) * 100

N-gram base similarity (s1, s2) = ((2 *16)/ (21+16))* 100
= (32/37) * 100
= 86.5%

Therefore we can say that the above two sentences are almost similar.

2.1.1.3. Word Order similarity:

Sentences containing the same words but in different orders may result in very different meanings. It is easy for humans to process word order information. However the incorporation of order information in to computational methods for understanding natural language is a difficult challenge. This may be the reason why most existing methods do not tackle this type of information. In this section we introduce a method that takes word order information into account when computing sentence similarity .For two sentences s1 and s2, r1 and r2 are the two arrays where r1 refers to array containing indexes of the words in the sentence s1 and r2 is an array containing indexes of the words in the sentences s2 w.r.t s1 then the word order similarity is calculated using the formulae

Word Order Similarity (s1, s2) =

$$(1 - (\sum_{i=0}^{n} |r1[i] - r2[i]| \ / \ \sum_{i=0}^{n} |r1[i] + r2[i]|))\text{X } 100$$

Example: The following example shows how the Word order similarity is calculated between a sample query and a sentence in a document.

Query: The dog jumps over a fox.
Sentence: The fox jumps over a dog.

Query is partitioned in to tokens: The |dog |jumps |over |a|fox.
After applying stop list: dog |jumps |fox.
Assigning the indexes: dog [0] |jumps [1] |fox [2].

Sentence is partitioned in to tokens: The |fox |jumps |over |a|dog.
After applying stop list: fox |jumps |dog.
Assigning the indexes w.r.t query: fox [2]| jumps [1] |dog[0]

Calculation of Word Order Similarity for above two sentences:

Word Order Similarity (s1, s2) =

(1 – (| 0 – 2 | + | 1 – 1 | + | 2 – 0 |) / ((0+2)+(1+1)+(2+0))) * 100

Word Order Similarity (s1, s2) = 1/3 * 100=33.4%.

Hence S1 and s2 are almost dissimilar

2.1.2.1. Semantic similarity:

Semantic similarity aims to find the similarity between two sentences based on the meaning of the individual words in the sentences. For a given query we create a set of related sentences by replacing an important word (i.e. noun, verb, adverb and adjective) by its synonym(s).so, the steps involved in calculating the semantic similarity are parts of speech tagging to the query, applying the stop list to remove insignificant words, finding the synonyms for every key word in a query, finally computing the similarity between the query synonyms and the sentences. So here for parts of speech tagging we use standard Stanford NLP POS tagger and for identifying the synonyms we use standard WordNet dictionary.

Example: The following example shows how the semantic similarity is calculated between a sample query and a sentence in a document.

Query: Computer is logical device.
Sentence: laptop is legitimate machine. `

Parts of speech tagging to the query:

Computer –noun is –verb logical–adjective device–noun. Query is partitioned in to tokens:
Computer –noun| is –verb| logical–adjective| device–noun.

After applying stop list:
Computer –noun| logical–adjective| device–noun.

Finding the synonyms for every key word in the sentence using WordNet Dictionary:

Computer: processor, CPU, mainframe,, laptop, pc Logical: rational, reasonable, sound, legitimate, valid.
Device: machine, tool, mechanism, gadget.

Sentence is partitioned in to tokens: laptop |is | legitimate|machine.
After applying stop list: laptop | legitimate |machine.

Finally finding the number of common key words between the words (keywords along with their synonyms) obtained from the query and the sentence.

Semantic similarity (s1, s2)=2* No Of Common Keywords in s1 & s2/ (total no of keywords in s1 and s2))
Semantic similarity (s1,s2)= 2*3/(3+3)=1

2.2 Sentence Clustering

After retrieving the important sentences by applying the proposed sentence scoring method, there is a need to check for redundancy among the sentences. Two or more sentences may convey same or similar meaning. One way to remove redundancy is though sentence clustering. In this paper clustering of sentences is done based on the clustering method proposed by Harish Karnicket. al [11]. The extracted sentences *S* are arranged in ascending order on the basis of score, then first sentence is selected and its similarity is measured with all the other sentence, sentences having similarity above the threshold are removed from the set S Similarly the procedure is repeated for all the sentences and then the outcome will be a summary without redundancy.

2.3 Sentence Ordering

To generate a readable coherent summary it is very important to order the sentences correctly. For a single document the order of sentence in the original document can be used as order to generate summaries. Alternately the sentences can be presented in descending order of their score [11].

3. PROPOSED SENTENCE SCORING METHOD

The proposed method finds relationship between a user query and a sentence in the text document. This method is defined in terms of the existing statistical and linguistic techniques. Sentences are extracted from the document based on the score given by the proposed sentence scoring method. These extracted sentences may contain redundant information and it is handled by using an iterative clustering algorithm. The final outcome contains the most important sentences in the document related to the query without redundancy which is known as summary of the document. The resultant sentences can be ordered according to their sentence score or order of sentences in the original document. The entire summarization process is shown in Figure 1.

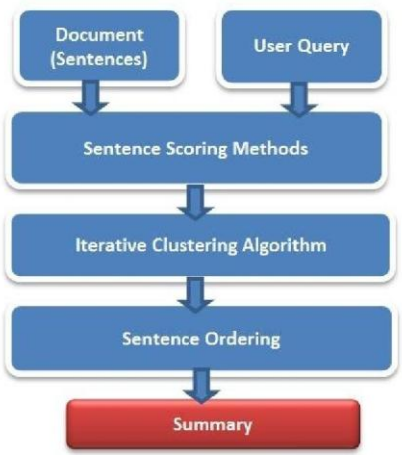

Figure 1. Summarization Process

Proposed sentence scoring method = ((a+b+c)/3+d)/2
where a, b, c and d refer to the similarities obtained by the sentence scoring methods Word form similarity, N-gram based similarity , Word Order Similarity and Semantic similarity respectively.

4. EVALUATION METRICS

There are many measures that can calculate the topical similarities between two summaries. For evaluation the results we use two methods. The first one is by precision (P), recall (R) and F1-measure which are widely used in Information Retrieval. For each document, the manually extracted sentences are considered as the reference summary (denoted by Summ_ref). In this approach the candidate summary (denoted by Summ_cand)is compared with the reference summary. The formulas for the measures P, R and F1 are as follows[9].Depending on what the summarization program focuses to make the summary of the text, summaries can be generic or query relevant. This paper mainly focused on query specific extractive summarization of source document. In extractive summarization each sentence in a document is scored based on some scoring method with respect to the query. The final outcome contains the most important sentences in the document related to the query without redundancy which is known as summary of the document. For single document order of sentence in the original document can be used as order to generate summaries.

$$P = (|\ summ\_ref \cap summ\_cand\ |)\ /\ summ\_cand$$

R = (| $summ\_ref \cap summ\_cand$ |)
/ $summ_ref$Error! Bookmark not defined.

F1= (2*P*R) / (P+R)

The second measure we use the ROUGE (Recall-Oriented Understudy for Gisting Evaluation), which was adopted by NIST (DUC) for evaluating automatic summarization system. It has been shown that ROUGE is very effective for measuring document summarization .It measures summary quality by counting overlapping units such as the N-gram, word sequences and word pairs between the candidate summary and the reference summary. The ROUGE-N [9] measure compares N-grams of two summaries, and counts the number of matches. The measure is defined by formula

ROUGE-N=

$$(\sum_{s\in summ\_ref} \sum_{N-grames} Count\_match(N-grams)) / (\sum_{s\in summ\_ref} \sum_{N-grames} Count(N-grams))$$

Where N stands for the length of the N-gram, Count_match (N-grams) is the maximum number of N-grams co-occurring in candidate summary and a set of reference summaries. Count (N-grams) is the number of N-grams in the reference summaries.

5. IMPLEMENTATION

The entire system is implemented in java programming language (J2SE). The following are the inputs and outputs of the developed system.

Input Document : Automatic summarization of a text document based on user query.doc

Query : what is automatic summarization of a text document Based on a user query?

Output (Summary):

Technology of automatic text summarization plays an important role in Information Retrieval Systems and Text Mining, and provides a solution to the information overload problem. Automatic text summarization is the creation of a shortened version of a text by a computer program. Automatic text summarization can be generic or query specific. This paper is focused on Query specific text summarization. Query specific text summarization is based on the calculation of the relationship between sentences in the text document and the query given. The technology of automatic text summarization is maturing and may provide a solution to the information overload problem. Automatic document summarization aims to condense the original text into essential content and to assist in filtering and selection of necessary information.

6. RESULT ANALYSIS

TABLE 1.Output in terms of no of sentences when each of the methods are applied individually on the source document consisting of 186 sentences

Sentence Scoring methods	Resulting No. of sentences from the source document
Word Form Similarity	3
N-gram Based Similarity	30
Word Order Similarity	7
Semantic Similarity	9
Proposed Similarity Measure	**12**

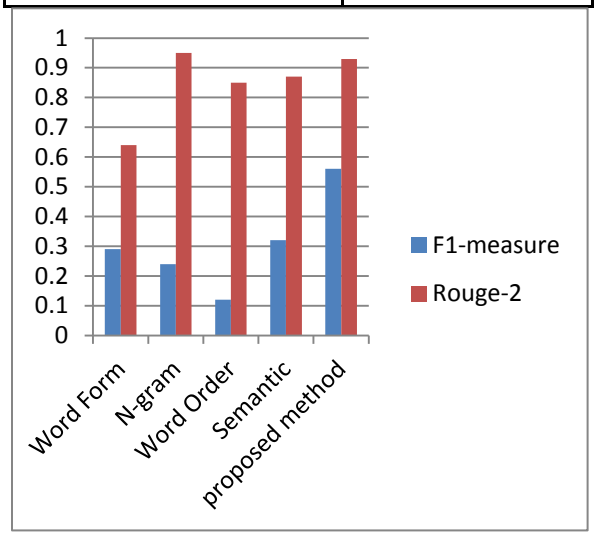

Figure 2: Graphical comparison of different sentence scoring methods

TABLE 2. Comparision of sentence scoring methods based on standard summary evaluation measures

Sentence Scoring Method	Precision	Recall	F1-Measure	ROUGE-2
Word Form Similarity	0.27	0.33	0.29	0.64
N-gram Based Similarity	0.16	0.55	0.24	0.95
Word Order Similarity	0.14	0.11	0.12	0.85
Semantic Similarity	0.33	0.33	0.32	0.87
Proposed Similarity	**0.50**	**0.66**	**0.56**	**0.93**

Note : The abstract of the document is taken as the reference summary to evaluate automatic summary.

7. CONCLUSION

It is observed from the results that the proposed sentence scoring method has given more accuracy in sentence scoring than the existing methods. Hence summary obtained is more relevant and close to manually generated summary.

8. FUTURE SCOPE

The proposed sentence scoring method is based on the average of the values scored using statistical techniques and linguistic techniques. Instead of just calculating average of the values, a weighted average can be taken where the weights for each of the values can be chosen based on the appropriateness of the methods applied for the content in the given document. Since a large size document will usually result in large no of relevant sentences the

clustering process can be implemented via a parallel processing technique like Threads in java to speed up the overall process of summarization.

9. REFERENCES

[1] Gholamrezazadeh, Saeedeh; Salehi, Mohsen Amini; Gholamzadeh, Bahareh, "A Comprehensive Survey on Text Summarization Systems," Computer Science and its Applications, 2009. CSA '09. 2nd International Conference on , vol., no., pp.1,6, 10-12 Dec. 2009
doi: 10.1109/CSA.2009.5404226

[2] YoungkoongKo, JungyunSeo, "An Effective Sentence-Extraction Technique Using Contextual Information and Statistical Approaches for Text Summarization", Pattern Recognition Letters. doi:10.1016/j.patrec.2008.02.008

[3] Wasson, M., "Using leading text for news summaries: Evaluation results and implications for commercial summarization applications", in Proc. 17th International Conference on Computational Linguistics and 36th Annual Meeting of the ACL, 1998, pp.1364-1368

[4] Salton, G, Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley Publishing Company, 1989.

[5] Waleed al-sanie, "Towards an infrastructure for Arabic text summarization using rhetorical structure theory", Master Thesis, Department of computer science. King Saud University, Riyadh, Kingdom of Saudi Arabia, 2005.

[6] http://web.science.mq.edu.au/~swan/summarization/projects_full.htm

[7] Bellegarda, J., "Exploiting latent semantic information in statistical language modeling," in Proc. IEEE, August 2000.Vol. 88, No. 8,pp: 1279-1296.

[8] R. L. Cilibrasi and P. M. B. Vitanyi, "The google similarity distance," IEEE Trans. On Knowl. and Data Eng., vol. 19, no. 3, pp. 370–383, 2007.

[9] Zhang Pei-ying and LI un-he,"Automatic text summarization based on sentences clustering and extraction" , IEEE2009

[10] Yuhua Li, Zuhair Bandar, David McLean and James O'Shea ,"A Method for Measuring Sentence Similarity and its application to conversational agents"

[11] Harish Karnick and VaruneshMishra,"Query Specific Multi-Document Summarization", Indian Institute of Technology, Kanpur, April 25, 2010

[12] Adamson, G and J. Boreham, " The use of an Association Measure Based on Character Structure to identify semantically related pairs ofwords and document titles".