Query-Based Summarizer Based on Similarity of Sentences and Word Frequency

Article · May 2011

DOI: 10.5121/ijdkp.2011.1301

P. Premchand
Osmania University

Dr A Govardhan
Jawaharlal Nehru Technological University, Hyderabad

Query-Based Summarizer Based on Similarity of Sentences and Word Frequency

A. P. Siva kumar[1], Dr. P. Premchand[2] and Dr. A. Govardhan[3]

[1] Department of Computer Science and Engineering, JNTUACE Anantapur, India
sivakumar.ap@gmail.com
[2] Professor, Department of Computer Science Engineering, Osmania University, Hyderabad, India
p.prechand@uceou.edu
[3] Principal & Professor, Department of Computer Science Engineering, JNTUHCE, Nachupalli, India.
govardhan_cse@yahoo.co.in

ABSTRACT

Text summarization is the most challenging task in information retrieval tasks. It is an outcome of electronic document explosion and can be seen as the condensation of the document collection. The use of text summarization allows a user to get a sense of the content of full-text, or to know its information content without reading all sentences within the full-text. Data reduction helps user to find the required information quickly without having to waste time in reading the whole text. We present a query based document summarizer based on similarity of sentences and word frequency. We used AQUAINT-2 Information-Retrieval Text Research Collections and the obtained summary sentences are evaluated using ROUGE metrics. The summarizer does not use any expensive linguistic data. Our Summarizer uses Vector Space Model for finding similar sentences to the query and Sum Focus to find word frequency, we achieved high Recall and Precision scores. The accuracy achieved using the proposed method is comparable to the best systems presented in recent academic competitions i.e., TAC (Text Analysis Conference).

KEYWORDS

Summarization, Sentence Similarity, Word Frequency, Query-based summarization.

1. INTRODUCTION

Information Retrieval (IR) is the science of searching for documents, information within documents, metadata about documents, relational databases and the World Wide Web. Summarization is a branch which deals with information retrieval.

Text summarization is the process of creating a summary of one or more text documents. For instance, we may summarize a large amount of news from different sources [1]. Many summarization techniques and their evaluation methods have been developed for this purpose. Such techniques are RANDOM [5], LEAD [5], MEAD [6] and PYTHY [9] etc. which are used to generate the summary. MEAD is the recent toolkit for summarization. We developed a multi-document, topic-driven summarizer. The input documents were newswire articles from AQUAINT-2 Information-Retrieval Text Research Collections and they were guaranteed to be related to their given topic. The topics themselves represent “real-world questions” that the summaries should answer. Two clusters of 10 articles, referred to as part A and part B, were assigned to each topic and a 100-word summary was created for each part.

DOI : 10.5121/ijdkp.2011.1301

Generic summarization processes have the following steps.

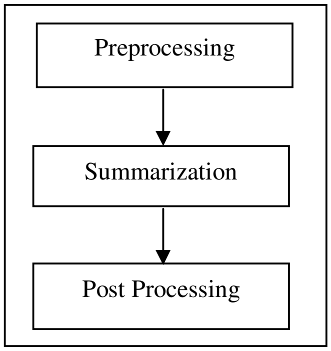

Figure 1. Steps in Summarization

Most of the existing summarizers produce summary which is redundant i.e. containing repeated information. In this paper we propose a query based summarizer which being based on grouping similar sentences and word frequency removes redundancy and has efficient Recall and Precision scores. Related work is presented in second section, system overview in the third, summary production in fourth, summary algorithm in fifth, evaluation and results in the sixth and future work and conclusions in seventh.

2.RELATED WORK

We present literature regarding summarization work based on grouping similar sentences and word frequency. Sum Basic uses term frequency as an approach to identify important sentences reducing information redundancy [2]. Local Topic Identification and word frequency are techniques used for Single Document Summarization [5]. Combination of other techniques with Similarity of first sentence for Multi Document Summarization [6].The use of frequency has proven useful in literature [3]. This is because authors state information in several ways [4]. We calculate similarity of sentences using cosine similarity measure [8]. Sum Focus is used to calculate word frequency [7]. After PreProcessing, producing the summary involves the following steps.

1. Calculate similarity of sentences present in documents with user query.
2. After calculating similarity group sentences based on their similarity values.
3. Calculate sentence score using word frequency and sentence location feature.
4. Pick the best scored sentences from each group and put it in summary.
5. Reduce summary length to exact 100 words.

3.SYSTEM OVERVIEW

The overview of our system is as shown in Fig.2. In the proposed system first the query is processed and the summarizer collects required documents matching with the summary and finally produces summary.

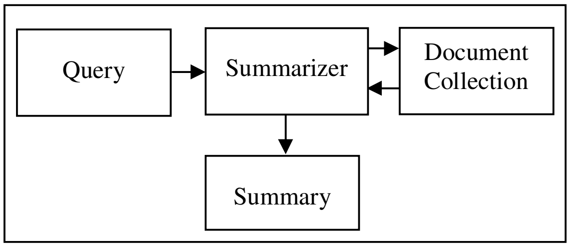

Figure 2. Overview of Summarization

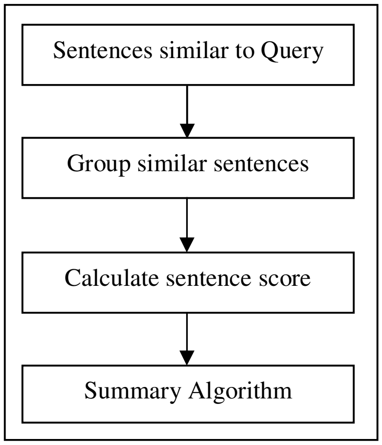

Figure 3. Stages in a Summarizer

The Figure.3. shows the description of summarization system used in the architecture of proposed system. Summarizer searches for the required document in document collection and performs the steps shown in the Figure 3.

4. SUMMARIZATION

4.1 SENTENCE SIMILARITY

Text summarization is a data reduction process. As summary is concise, accurate and explicit, it has great significance.

Sentence similarity means we calculate how much they are similar. Sentence similarity is calculated by most widely used Vector Space Model (VSM). VSM model classifies the sentences based on calculated similarity value.

4.1.1. VSM MODEL

In Vector Space Model both the sentences in document and query are arranged in vectors. In the late 1958, Luhn [12] first suggested that automatic text retrieval systems could be designed based on a comparison of content identifiers attached both to the stored texts and to the user's information queries. Certain words extracted from the texts of documents and queries would be used for content identification. In either case, the documents would be represented by *term vectors* of the form

$$D = (Wt1, Wt2, \dots\dots. Wtk)$$

Where each W_{t1}, W_{t2}…..W_{tk} identifies a content term assigned to some sample document D. Analogously, the information requests, or queries, would be represented either in vector form, or in the form of Boolean statements.

Thus, a typical query Q might be formulated as

$$Q = (Wq1, Wq2, \dots\dots. Wq3)$$

Where W_{q1}, W_{q2} ….. W_{qk} represents weights of content term in query.

W_{t1} - Weight of content term in a Sentence,

W_{q1} – Weight of content term in a Query.

4.2. TERM WEIGHT

4.2.1. TERM FREQUENCY (TF)

Number of times a term occurred in a sentence is called as the 'Term frequency'. It is represented as "tf".

4.2.2. DOCUMENT FREQUENCY (DF)

Number of times a term occurred in the whole document is called as 'Document frequency'. It is represented as "df".

4.3.3. INVERSE DOCUMENT FREQUENCY (IDF)

It is the logarithm of inverse of the document frequency. Inverse document frequency is represented as "idf".

$$idf = \log\left(\frac{n}{df}\right)$$

where n – Number of sentences

df – document frequency

Term weight

Term Weight is a measure used to calculate weight of term which is scalar product of term frequency and inverse document frequency mathematically represented as follows.

$$W = tf * idf$$

We calculate W_{t1}, W_{t2} …W_{tk} which are word weights of content words in documents. W_{q1}, W_{q2} ……..W_{qk} are word weights of content terms in query.

We calculate W_{ti}, W_{qi} which are indexing terms given to a document. The similarity is calculated by widely used cosine measure.

$$Sim\,(Q,D) = \frac{\sum_{i=1}^{t} Wqi * Wdi}{\sqrt{\sum_{i=1}^{t}(Wqi)^2 * \sum_{i=1}^{t}(Wdi)^2}} \qquad (1)$$

With the above measure we get similarity values which can be stored for further processing.

4.3. GROUP SIMILAR SENTENCES

After finding the similarity of sentences we arrange the sentences in a particular order. i.e., based on the similarity values obtained. We arrange the sentences in ascending order based on similarity values. Then we form a group based on their similarity. The sentences which have same similarity value form a particular group which has its associated similarity value.

4.4. WORD WEIGHTS

After forming groups we have to compute and order each group, pick in a cyclic fashion the best sentence from the best group if the desired summary length has not been reached. The core system is Sum Basic. Sum Basic is a generic algorithm; it does not include other features or information, therefore, we improved the Sum Basic by sentence location feature and Sum Focus method.

4.3.1. SUM FOCUS

Sum Focus made by Lucy Vanderwende [7], a new approach in the multi-document summarization system, captures the information conveyed by the topic description by computing the word probabilities of the topic description. The weight for each word is computed as a linear combination of the unigram probabilities derived from the topic description, with back off smoothing to assign words not appearing in the topic a very small probability, and the unigram probabilities from the document, in the following manner:

The weight of each word is computed as a linear combination of unigram probabilities derived from the description.

$$Word\ Weight = DocWt * GrpWt \qquad (2)$$

DocWt represents weight of a word in whole document whereas GrpWt represents weight of word in a group. The scalar product of document and group weights results in word weight of a word. Word weights of each word are useful while producing the final summary.

4.4. SENTENCE SCORE

Sentence score is a measure which is used to measure how important the sentence is in document. We calculate the sentence score of a sentence by using the following measure.

$$Sentence(Sj) = \sum_{Wi \epsilon Si} \frac{Word\ Weight}{(|\{Wi|Wi \epsilon Sj\}|)} \qquad (3)$$

- Word weight is represented in (2).
- W_i represents the word number in a sentence.
- S_i represents the sentence position in a document.

Individual Sentence score is calculated for the finding the score of the entire group.

4.5. SENTENCE LOCATION FEATURE

Sentence Location Feature is also important except for the words occurring frequently.

This is a feature which is used to adjust the weight of sentences and is measured as follows.

$$Location\ (Lj) = (1 + \frac{(p-0.75*m)^2}{m^2} * STj) \qquad (4)$$

For each sentence L_j in group

P – Serial number of sentences

M – Number of sentences in document

ST_j – Similarity values between sentence and query

For each sentence L_j in the group P is the serial number of sentence L_j in the document. 'm' is the number of sentence in the document. ST_j is the similarity value between sentence L_j and title of the document.

4.6. GROUP SCORE

For each sentence S_j in the groups, compute the Sentence (S_j) and Location (L_j), calculate the score of each sentence and the score of each group. We used following algorithm

$$Score(Grp) = \sum\big(\alpha * Sentence(Sj) + \beta * Location(Lj)\big) \qquad (5)$$

Sentence (S_j) – Sentence Score represented in (3)

Location (L_j)–Sentence Location Feature represented in (4)

4.7. SUMMARY ALGORITHM

To produce the summary, we propose the following algorithm:

Step 1:

Compute the word weight of W_i words appearing in a document using formula (2).

Step 2:

Compute the Sentence score (S_j) using formula (3) and Location (L_j) using formula (4).

Step 3:

Using the formula (5) calculate the score of each group.

Step 4:

Arrange the groups in ascending order depending on their group scores.

Step 5:

From the best group, pick the sentence having the maximum sentence score.

Step 6:

Delete the group and repeat *Step 5* until all the sentences are picked from each group.

Step 7:

If length of a summary is greater than desired then shrink the summary to required length.

5. EVALUATION AND RESULTS

To evaluate our summarization system we used TAC2009 datasets proposed by NIST for update summarization task. We conducted all the experiments on TAC 2009 Update Summarization dataset. It consists of 48 topics, each having 20 documents divided into two clusters "A" and "B" based on their chronological coverage of topic. It serves as an ideal setting for evaluating our Multidocument summaries. Summary for cluster A (pdocs) is a normal multi document summary where as summary for cluster B (ndocs) is a Progressive summary, both of length 100 words. Each topic has associated 4 model summaries written by human assessors. Summary can be evaluated using N-gram Co-occurrences [11].

5.1 ROUGE EVALUATION

Summaries are evaluated using ROUGE [10], a recall oriented metric that automatically assess machine generated summaries based on their overlap with models. ROUGE-2 and ROUGE-SU4 are standard measures for automated summary evaluation. For evaluation of our system we used cluster 'A' documents of TAC2009 data. In cluster A we tested 15 topics of datasets and evaluated with ROUGE metrics.

The Table 1. shows the evaluation results of our system. For evaluation we used 15 topics present in the TAC2009 data along with the associated documents to test our system. The average score for all the topics tested with our system is displayed in the system with ROUGE

Table 1. Average recall, precision, F-score values of our system

	Avg_R	Avg_P	Avg_F
ROUGE-1	0.30127	0.29034	0.29961
ROUGE-2	0.049802	0.048922	0.048994
ROUGE-L	0.13374	0.12997	0.13215
ROUGE-SU4	0.08225	0.08017	0.08132

The Table 1. represents the average recall, average precision, and average F-score generated by ROUGE package for our system.

5.2 COMPARISON OF RECALL VALUES

We compare the average Recall values of our system with TAC2009 participants. We calculated average Recall values of participants using the Evaluation Results of TAC2009. Average Recall is calculated for the 15 topics in our testing part. The Table 2. gives the comparison for ROUGE-1, 2, L, SU4 metrics.

Table 2. Comparison of average Recall of our system with participants of TAC2009

	Avg_R (Participants)	Avg_R (Our System)
ROUGE-1	0.334473	0.30127
ROUGE-2	0.081579	0.049802
ROUGE-L	0.205363	0.13374
ROUGE-SU4	0.115487	0.08225

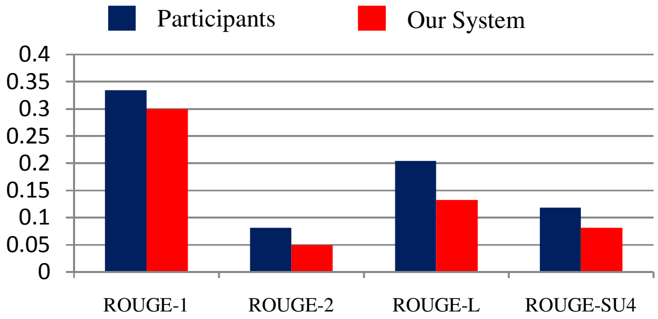

Figure 4. Comparison of Average Recall values for 4 metrics of ROUGE.

5.3. COMPARISON OF PRECISION VALUES

We compare the average Precision values of our system with those of TAC2009 participants. We calculated average Precision values of participants using the Evaluation Results of TAC2009. Average Precision is calculated for the 15 topics in our testing part. The Table 3. gives the comparison for ROUGE-1, 2, L, SU4 metrics.

Table 3. Comparison of average Precision of our system with participants of TAC2009

	Avg_P (Participants)	Avg_P (Our System)
ROUGE-1	0.338458	0.29034
ROUGE-2	0.081509	0.048922
ROUGE-L	0.209295	0.12997
ROUGE-SU4	0.122135	0.08017

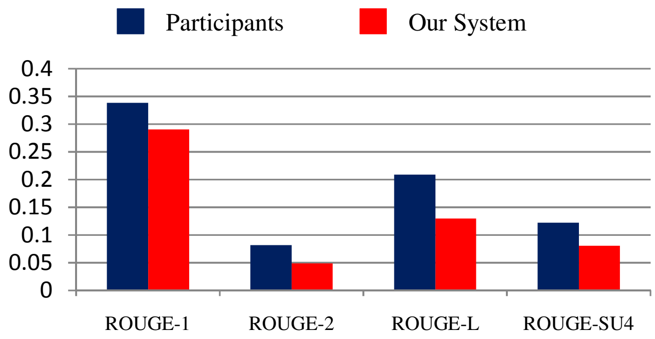

Figure 5. Comparisons of Average Precision values for 4 metrics of ROUGE.

5.4. COMPARISON OF F-SCORE VALUES

We compare the average F-score values of our system with TAC2009 participants. We calculated average F-score values of participants using the Evaluation Results of TAC2009. Average F-score is calculated for the 15 topics in our testing part. The Table 4. shows the comparison for ROUGE-1, 2, L, SU4 metrics.

Table 4. Comparison of average F-Score of our system with participants of TAC2009

	Avg_F (Participants)	Avg_F (Our System)
ROUGE-1	0.334395	0.29961
ROUGE-2	0.081098	0.048994
ROUGE-L	0.203905	0.13215
ROUGE-SU4	0.118196	0.08132

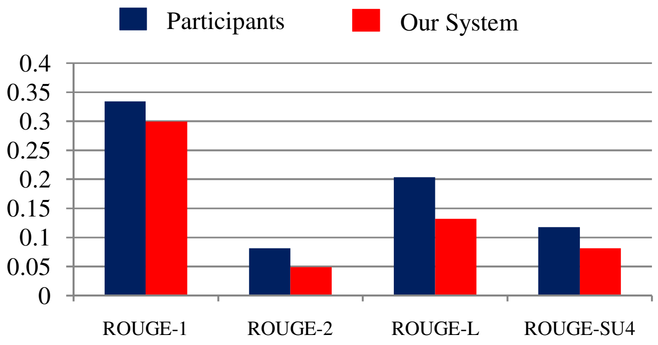

Figure 6. Comparisons of Average F-score values for 4 metrics of ROUGE

6.FUTURE WORK AND CONCLUSION

In this paper we propose an approach for query-based summarizer based on grouping similar sentences and word frequency. Experimental results demonstrate that our system achieves the best average recall, precision, and F-score.

In future, we would like to improve the system by adding sentence simplification techniques for producing summary. We can add sentence simplification feature to simplify the sentences which are complex and very large. With the implementation of sentence simplification, more informative content can be added in summary by creating more space to add sentences which increases linguistic quality and readability to a large extent.

We evaluated our work with TAC2009 datasets and evaluation results are presented in evaluation section.

REFERENCES

[1] Radev, D.R., Jing, H., & Budzikowska, M. (2000). Centroid - based summarization of multiple documents: sentence extraction, utility based evaluation, and user studies. In the proceedings of the NAACL/ANLP Workshop on Automatic Summarization (pp. 21-30), Seattle, Washington: ACL.

[2] Nenkova, A., & Vanderwende, L. (2005). The impact of frequency on summarization. No. MSR-TR-2005-101. Redmond, Washington: Pergamon Press.

[3] Hovy, E., & Lin, C. (1999). Automated text summarization in SUMMARIST. *In I. Mani & M. T. Maybury (Eds.),* Advances in Automatic Text Summarization (pp. 81-94). Cambridge, MA: MIT Press.

[4] Sparck Jones, K. (1999). Automatic summarizing: Factors and directions. In I. Mani & M. T. Maybury (Eds.), Advances in automatic text summarization (pp. 2-12). Cambridge, MA: MIT Press.

[5] Zhi Teng., Ye Liu., Fuji Ren ., Seiji Tsuchiya., and Fuji Ren "Single Document Summarization Based on Local Topic Identification and Word Frequency" In Seventh Mexican International Conference on Artificial Intelligence 2008.

[6] Md. Mohsin Ali ., Monotosh Kumar Ghosh., and Abdullah-Al-Mamun., " Multi-document Text Summarization: SimWithFirst Based Features and Sentence Co-selection Based Evaluation " In International Conference on Future Computer and Communication 2009.

[7] Khan, A.U.,Khan, S., & Mahmood, W. (2005). MRST: A new technique for information summarization. In the Proceedings of Second World Enformatika Conference (pp. 249-252), Istanbul, Turkey: (WEC'05).

[8] Radev, D.R., Teufel, S., Saggion, H., Lam, W., Blitzer, J., Qi, H, "et al". (2003). Evaluation challenges in large-scale multi-document summarization: the mead project. In the Proceedings of ACL, Sapporo, Japan: ACL.

[9] Lucy Vanderwende, Hisami Suzuki, Chris Brockett and Ani Nenkova. (2007). Beyond Sum Basic: Task-focused summarization with sentence simplification and lexical expansion (pp. 1606-1618). *Information Processing and Management*, 43 2007 volume 6 November 2007, Tarrytown, NY, USA: Pergamon Press.

[10] Harman, D. K. (1995). Overview of the fourth text retrieval conference (TREC-4). In D. K. Harman (Ed.), Proceedings of the fourth text retrieval conference (pp. 1-24). *NIST Special Publication 1996.*

[11] Kristina Toutanova, Chris Brockett, Michael Gamon, Jagadeesh Jagarlamundi, Hisami Suzuki, and Lucy Vanderwende. (2007). the pythy summarization system: Microsoft research at DUC 2007. In the proceedings of Document Understanding Conference.

[12] Chin-Yew Lin. (2004) "ROUGE : A Package for Automatic Evaluation of Summaries", In the Proceedings of the Workshop on the Text Summarization Branches Out (WAS 2004), Barcelona, Spain: ACL.

[13] Chin-Yew Lin, and Edward Hovy, (2003) "Automatic Evaluation of Summaries Using N-gram Co-occurrences Statistics", In Proceedings of 2003 Language Technology Conference (HLT-NSSCL, 2003), Edmonton, Canada, 2003.

[14] Luhn. HP. (1958) "The Automatic Creation of Literature Abstracts", IBM Journal of Research and Development (pp. 159-165).

[15] Katragadda, R., Pingali, P., and Varma, V., "Sentence position revisited: A robust light-weight update summarization 'baseline' algorithm", In Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies (CLIAWS3), pages 46–52, Boulder, Colorado. Association for Computational Linguistics.

[16] Li, J., Sun, L., Kit, C., and Webster, J., "A query-focused Multidocument summarizer based on lexical chains", In DUC'07: Document Understanding Conference, 2007.

[17] Bysani, P., Bharat, V., and Varma, V., "Modeling novelty and feature combination using support vector regression for update summarization", In 7th International Conference On Natural Language Processing. NLP Association of India

[18] Berger, A. and Mittal, V. O., "Query-relevant summarization using faqs", In ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pages 294–301, Morristown, NJ, USA. Association for Computational Linguistics.

[19] Kupiec, J., Pedersen, J., and Chen, F. , "A trainable document summarizer", In the proceedings of ACM SIGIR'95, pages 68–73. ACM.

[20] Conroy, J. M., Schlesinger, J. D., Goldstein, J., and O'leary, D. P., " Left-brain/right-brain multi-document summarization", In the proceedings of Document Understanding Conference (DUC) 2004.

[21] Shen, D., Sun, J.-T., Li, H., Yang, Q., and Chen, Z., "Document summarization using conditional random fields", In the proceedings of IJCAI '07., pages 2862–2867. IJCAI.

[22] Lin, C.-Y. , "Rouge: A package for automatic evaluation of summaries", In the proceedings of ACL Workshop on Text Summarization Branches Out. ACL.

[23] Jagarlamudi, J., "Query-based multi-document summarization using language" Master's report, IIIT Hyderabad, India.

[24] Allan, J., Gupta, R., and Khandelwal, V., "Topic models for summarizing novelty", In Proceedings of the Workshop on Language Modeling and Information Retrieval, pages 66–71.

[25] Md. Mohsin Ali ., Monotosh Kumar Ghosh., and Abdullah-Al-Mamun., “Multi-document Text Summarization: SimWithFirst Based Features and Sentence Co-selection Based Evaluation” In International Conference on Future Computer and Communication 2009.

[26] Zhi Teng., Ye Liu., Fuji Ren ., Seiji Tsuchiya., and Fuji Ren “Single Document Summarization Based on Local Topic Identification and Word Frequency” In Seventh Mexican International Conference on Artificial Intelligence 2008.