Cover Coefficient-Based Multi-document Summarization

Gonenc Ercan and Fazli Can

Computer Engineering Department, Bilkent University
Ankara, Turkey

Abstract. In this paper we present a generic, language independent multi-document summarization system forming extracts using the cover coefficient concept. Cover Coefficient-based Summarizer (CCS) uses similarity between sentences to determine representative sentences. Experiments indicate that CCS is an efficient algorithm that is able to generate quality summaries online.

Keywords: Multi-document Summarization, Cover Coefficient Concept, Automated Text Summarization.

1 Introduction

In this paper we attack the problem of forming an extract for a set of documents about a single topic. It is possible to appreciate the importance of such a task only by considering its applications. News portals can provide precise summaries about a news merged from multiple source articles.

Most of the current summarization systems consider running time of the algorithms as a reasonable tradeoff for the quality of the summaries generated, since in most of the applications the summaries are generated offline. However in emerging applications such as Vivisimo's Clusty search engine[1] may require online generation of summaries. Such a search engine can present short summaries of each cluster for a better browsing experience. Most of the current summarization algorithms are not suitable for such applications, as they are demanding and language dependent. CCS algorithm can prove to be useful in such applications as it is language independent, efficient and achieves competitive ROUGE scores when compared to state of the art summarization systems.

The contributions of this paper are the development of a language independent multi-document summarization algorithm that uses a double-stage probability experiment to determine the most significant sentences and checking for repetition with a Boolean function that derives a similarity threshold for a pair of sentences from the whole document set with a constant number of cover coefficient (CC) calculations as explained in Section 2.

An ideal summarization system, must interpret the text, which requires extensive processing of the text. Important portion of the research on summarization

[1] www.clusty.com

M. Boughanem et al. (Eds.): ECIR 2009, LNCS 5478, pp. 670–674, 2009.

uses deeper levels of language modelling [1,2]. Some research uses ideas from information retrieval for summarization. Radev et al. [3] uses sentence level vector space model, to identify sentences that are most similar to the centroid. This algorithm is extended by introducing a prestige factor to sentences [4]. Avoiding repetition in summarization has been addressed both by Radev et al. [3] and Carbonell et al. [5].

2 CC-Based Multi-document Summarizer

CC concept is first introduced for clustering documents [6], using a document by term matrix. The term document is flexible, such that it is possible to replace it with sentences or any other text chunk representable as a bag of words such as paragraphs. In CC the S matrix is transformed into a sentence by sentence CC matrix denoted by C, where S matrix is composed of sentence term occurence vectors. Each element in C, such as c_{ij} can be read as how much s_j covers s_i. Elements of the C matrix is calculated by using a double-stage probability experiment.

$$c_{ij} = \sum_{k}^{n} \alpha_{ik} * \beta_{kj} \qquad 1 \leq i, j \leq m \tag{1}$$

Equation 1 is the calculation of the probability c_{ij}, which defines coverage probability as the joint probabilities of α and β probabilities. Let n denote the number of terms and m denote the number of sentences. The α_{ik} probability is the probability of selecting term k from sentence i. The term β_{ki} is the probability of term k occurring in sentence i.[2]

Since all of these probabilities constitute the whole probability space, sum of all c_{ij} values for a sentence i is equal to 1. With this fact we can immediately assume that c_{ii} values are the dissimilarity of sentence i to (decoupling from) other sentences. From the other way around it is possible to say that $1 - c_{ii}$ is how much sentence i is covered by other sentences. As these two values are of great value we will denote them with δ_i and Ψ_i symbols respectively [6].

It is beneficial to present a complete example of the CCS algorithm using the example S matrix shown in Figure 1(a). Figure 2 shows the coverage probability graph of s_1. Sum of all paths from s_1 to s_2 shows how probably s_2 covers s_1, which we refer to as c_{12}. Figure 1(b) shows the resulting CC matrix.

$$S = \begin{vmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{vmatrix} \qquad C = \begin{vmatrix} \mathbf{0.42} & 0.25 & 0.17 & 0.00 & 0.17 \\ 0.17 & \mathbf{0.44} & 0.00 & 0.28 & 0.11 \\ 0.17 & 0.00 & \mathbf{0.42} & 0.00 & 0.42 \\ 0.00 & 0.42 & 0.00 & \mathbf{0.42} & 0.17 \\ 0.11 & 0.11 & 0.28 & 0.11 & \mathbf{0.39} \end{vmatrix}$$

(a) Example S Matrix (b) Cover Coeficient Matrix

Fig. 1. Example Matrices

[2] Note that this equation and definitions differ from [6,9].

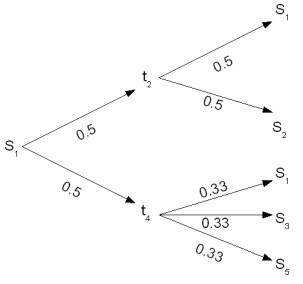

Fig. 2. Probability graph of s_1

All the similarity values for sentences (Ψ_i) are calculated, and sorted. Top sentences are the most central sentences, and thus should be included in the target summary. Avoiding repetition in the summary is a problem that must be addressed in multi-document summarization. This problem can be solved by selecting only candidate sentences that are not covered by an already selected sentence. This can be considered as checking how novel a candidate sentence is. The probability of s_j covering s_i is the c_{ij} value, where s_j is an already selected summary sentence and s_i is a candidate sentence considered for inclusion. The problem is determining if c_{ij} probability is too high, indicating a repetition. The diagonal value of s_i is the c_{ii}, which is the coverage probability of s_i covering itself. Since s_i is a perfect cover of itself, its value can be used in a decider for repetition. Our criterion for repetition is $c_{ij} > \frac{c_{ii}}{\mu}$ or $c_{ji} > \frac{c_{jj}}{\mu}$, where μ is a constant value. Setting μ value to 2, is analogous to deciding that there is a repetition if the coverage probability is greater than half of the perfect cover's coverage probability. We have seen experimentally that setting the μ value to 4, achieves the best results.

The coverage probability unlike similarity, is not symmetric. Figure 2 shows two sentences from Duc 2004 corpus detected by our algorithm to be repeating the same information. The probability of s_2 covering s_1 is c_{12}, and probability of s_1 covering s_2 is c_{21}. These two values are not the same as s_2 presents extra information not available in s_1. In our implementation, both of the probabilities are checked for repetition.

Continuing with our example, sentence s_5 is selected to the summary, as Ψ_5 is highest. There are 3 sentences with 0.58 in our example, in this case our algorithm chooses a random sentence from these sentences. Perfect cover of sentence s_1 is 0.42, and c_{15}, c_{51} values can be calculated as 0.17 and 0.11 respectively. When the μ value is set to 2, sentence s_1 is not a repetition of s_5, and included in the summary. Next candidate sentence is s_3, c_{35} and c_{53} is 0.42 and 0.28 respectively. Sentence s_3 is a repetition of s_5, so it is not included in the summary. This process is repeated until there are no more candidate sentences left or the target summary size is reached.

s_1: `On Saturday, the rebels shot down a Congolese Boeing 727 over the Kindu airport.`
s_2: `On Saturday, the rebels said they shot down a Congolese Boeing 727 which was attempting to land at Kindu air base with 40 troops and ammunition.`

Fig. 3. Repeating Sentences

3 Experimental Results

Document Understanding Conference [7] has been a testbed for automated summarization research for over a decade. DUC 2004 corpus consists of 50 topics, each containing 10 related news articles. For evaluation purposes four human annotators have summarized each topic, so that each system can evaluate their abstracts by comparing it with the manually created summaries. For the multi-document summarization task, the target size is 665 characters.

Table 1. DUC2004 Task 2 Corpus Results using ROUGE

Score Type	Systems			
	CCS	**MEAD**	**Avg.**	**Best**
ROUGE-1	0.376(2)	0.348(16)	0.339	0.382
ROUGE-2	0.082(8)	0.073(20)	0.069	0.092
ROUGE-3	0.025(13)	0.024(20)	0.022	0.035
ROUGE-L	0.339(1)	0.275(27)	0.293	0.333
ROUGE-W	0.118(1)	0.110(27)	0.102	0.116

ROUGE [8] is commonly used for summarization evaluation. ROUGE compares system summaries with manually created summaries. Comparison is done by different metrics such as N-Grams and Longest Common Subsequences (LCS). In Table 1 the ROUGE scores for CCS is given. ROUGE-N denotes N-Gram based similarities from 1-grams to 3-Grams. ROUGE-L denotes LCS and ROUGE-W denotes weighted LCS. In DUC2004 there were 35 systems that participated in multi-document summarization task. For comparison the average and best scores are given. MEAD [3] summarization toolkit also participated in DUC2004. Their algorithm uses centroid feature combined with position in text and LexRank score [4]. Centroid feature used by MEAD takes advantage of the lexical centrality of sentences, so it is reasonable to compare our algorithm with theirs. The ranks of the systems are given in parentheses.

CCS ranked 2^{nd} in ROUGE-1 score. In ROUGE-2 and ROUGE-3 scores, CCS achieved lower ranks than the ROUGE-1 score. Our system achieves the best ROUGE-L and ROUGE-W scores among 35 systems.

4 Conclusion and Future Work

CCS algorithm is a novel technique for multi-document summarization, that could be used in online generation of summaries in emerging applications. The results are promising as, the algorithm achieves competitive results when compared to 35 other state of the art systems and surface level language processing is adequate.

In our evaluations, we were not able to show the effectiveness of the Boolean repetition check function. ROUGE does not directly evaluate repetition in the summary, thus a new evaluation technique should be used. An attempt for single

document summarization could yield good results. Currently only CC values are used in the summarizer, however there are features such as sentence position in text and temporal features that are used with success in summarization. We are in the process of integrating these features. With our motivations in using CCS in search engines with document clusters, it could be reasonable to compare the running time of our algorithm with snippet algorithms for search engines. Algorithm can be extended to support incremental summarization for dynamic set of documents that may change in time, using the ideas from incremental clustering [9]. For example, news and event tracking systems may benefit from this approach to generate summaries for events on the fly.

Acknowledgements

This work is partially supported by The Scientific and Technical Council of Turkey Grant "TUBITAK EEEAG-107E151" and "TUBITAK EEEAG-108E074".

References

1. Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: Proceedings of the ACL/EACL, pp. 10–17 (1997)
2. Marcu, D.: From discourse structures to text summaries. In: Proceedings of the ACL/EACL, pp. 82–88 (1997)
3. Radev, D., Jing, H., Budzikowska, M.: Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In: Proceedings of NAACL-ANLP, pp. 919–938 (2000)
4. Gunes, E., Radev, D.R.: LexRank: graph-based centrality as salience in text summarization. Journal of Artificial Intelligence Research (JAIR) 22, 457–479 (2004)
5. Carbonell, J.G., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of Special Interest Group of Information Retrieval, pp. 335–336 (1998)
6. Can, F., Ozkarahan, E.A.: Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases. ACM Transactions on Database Systems 15(4), 483–517 (1990)
7. Document Understanding Conference, `http://duc.nist.gov`
8. Lin, C., Hovy, E.H.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003), pp. 71–78 (2003)
9. Can, F.: Incremental clustering for dynamic information processing. ACM Transactions on Information Systems 11(2), 143–164 (1993)