Multi-document Summarization Based on BE-Vector Clustering

Dexi Liu[1,2,3], Yanxiang He[1,3], Donghong Ji[3,4], and Hua Yang[1,3]

[1] School of Computer, Wuhan University, Wuhan 430079, P.R. China
[2] School of Physics, Xiangfan University, Xiangfan 441053, P.R. China
[3] Center for Study of Language and Information, Wuhan University,
Wuhan 430079, P.R. China
[4] Institute for Infocomm Research, Heng Mui Keng Terrace 119613, Singapore
dexiliu@gmail.com, yxhe@whu.edu.cn,
dhji@i2r.a-star.edu.sg, yh@eis.whu.edu.cn

Abstract. In this paper, we propose a novel multi-document summarization strategy based on Basic Element (BE) vector clustering. In this strategy, sentences are represented by BE vectors instead of word or term vectors before clustering. BE is a head-modifier-relation triple representation of sentence content, and it is more precise to use BE as semantic unit than to use word. The BE-vector clustering is realized by adopting the k-means clustering method, and a novel clustering analysis method is employed to automatically detect the number of clusters, K. The experimental results indicate a superiority of the proposed strategy over the traditional summarization strategy based on word vector clustering. The summaries generated by the proposed strategy achieve a ROUGE-1 score of 0.37291 that is better than those generated by traditional strategy (at 0.36936) on DUC04 task-2.

1 Introduction

With the rapid growth of online information, it becomes more and more important to find and describe textual information effectively. Typical information retrieval (IR) systems have two steps: the first is to find documents based on the user's query, and the second is to rank relevant documents and present them to users based on their relevance to the query. Then the users have to read all of these documents. The problem is that these docs are much relevant and reading them all is time-consuming and unnecessary. Multi-document summarization aims at extracting major information from multiple documents and has become a hot topic in NLP. Multi-document summarization can be classified into three categories according to the way that summaries are created: sentence extraction, sentence compression and information fusion.

The sentence extraction strategy ranks and extracts representative sentences from the multiple documents. Radev [1] described an extractive multi-document summarizer which extracts a summary from multiple documents based on the document cluster centroids. To enhance the coherence of summaries, Hardy Hilda [2] and Mitra [3] extracted paragraphs instead of individual sentences.

A. Gelbukh (Ed.): CICLing 2006, LNCS 3878, pp. 470 – 479, 2006.

Knight and Marcu [4] introduced two algorithms for sentence compression based on the noisy-channel model and the decision-tree approach. The input to each algorithm is the parse tree of a long sentence, and the output is expected to be a reduced sentence keeping the major semantic information. However, it is hard to control the compression ratio using this strategy.

Barzilay [5] described an algorithm for information fusion, which tries to combine similar sentences across documents to create new sentences based on language generation technologies. Although this strategy can simulate, to some degree, the human's action in summarization process, it heavily relies on some external resources, e.g. dependency parsers, interpretation or generation rules, etc, which inevitably limit its portability.

In the sentence extraction strategy, clustering is frequently used to eliminate the redundant information resulted from the multiplicity of the original documents [6]. There are two levels of clustering granularity: sentence and paragraph. Generally, word is employed as the minimal element of a document [1]. However, word may be not precise enough for clustering. So the researchers have turned to terms as the semantic unit [7]. The trouble is that most term extraction methods are based on statistical strategy, thus, a term is not a real syntactic or semantic unit.

In this paper, we apply Basic Elements (BE) [8] as the minimal semantic unit. BE is a head-modifier-relation triple representation of document contents developed for summarization evaluation system at ISI, and is intended to represent the high-informative unigrams, bigrams, and longer units of a text, which can be built up compositionally. BEs can be generated automatically without the support of large corpus that terms based on.

This multi-document summarization approach (MSBEC for abbreviation) consists of four main stages: 1) Preprocessing: break down sentences into BEs and calculate the score of each BE and each sentence. 2) BE clustering: represent each sentence with a BE-vector and apply the k-means clustering method on these BE-vectors. 3) Sentence selection: from each cluster, select a sentence with highest score as the representation of this cluster. 4) Summary generation: output the selected sentences to form the final summary according to their positions in the original documents.

We also propose a novel clustering analysis method, which is based on evaluating the cohesion of within-clusters and the scatter of between-clusters, to automatically determine *K*, the number clusters.

The rest of this paper is organized as follows. In the next section, we give a short overview of Basic Elements. Section 3 describes the strategy of multi-document summarization based on BE-vector clustering. Section 4 shows the performance comparison of BE-vector clustering and word-vector clustering. Finally, we conclude this paper and discuss future directions in Section 5.

2 Basic Element

Basic Element [8] is a relation between a head-BE and a single dependent, expressed as a triple (head | modifier | relation), where "head" denotes the head of a major syntactic constituent (noun, verb, adjective or adverbial phrases).

Figure 1 presents BE examples for "The United Nations imposed sanctions on Libya in 1992 because of their refusal to surrender the suspects". BEs can be extracted automatically in several ways. Most of them use a syntactic parser to produce a parse tree and then apply a set of ‘cutting rules’ to extract valid BEs from the tree. In this paper, we use the BE package 1.0 [8] distributed by ISI.

With the triple BE, one can quite easily decide whether any two units match (express the same meaning), and word in BEs is more meaningful. For instance, “United Nations”, “UN”, and “UNO” can be matched at this level (but require work to isolate within a longer unit or a sentence), allowing any larger unit encompassing this to accept any of the three variants. Moreover, the pronoun “their” in the example sentence designates “United Nations” clearly.

head	**modifier**	**relation**	
imposed	united nations	subj	(BE-F)
imposed	sanctions	obj	(BE-F)
sanctions	libya	on	(BE-F)
libya	1992	in	(BE-F)
refusal	their	gen	(BE-F)
libya	refusal	because of	(BE-F)
refusal	surrende	comp1	(BE-F)
surrender	united nations	subj	(BE-F)
surrender	suspects	obj	(BE-F)

Fig. 1. Example of BEs in a sentence: “The United Nations imposed sanctions on Libya in 1992 because of their refusal to surrender the suspects.”

3 Multi-document Summarization Based on Basic Elements

In this section, we will introduce the Basic Element-based summarization strategy in details.

3.1 Preprocessing

The preprocessing stage consists of 3 sub-steps.

3.1.1 BE Generation

To break down sentences in a document set into BEs, we employ the BE breaker module in the BE Package distributed by USC/ISI. This module first uses the Minipar [9] parser to create the syntactic tree and then prune it. Once relations between its nodes are resolved, it can result in a list of BEs illustrated in figure 1.

3.1.2 BE Score Calculation

To distinguish which BE is indeed important and uniquely indicative in the document set, we calculate for each BE its informativeness. Every BE has three parts: head-BE, modifier and the relation between head and its modifier, where the head-BE is more representative for the meaning of a BE than the other parts. So the calculation of BE score is replaced by the calculation of head-BE score. We adopt the typical word weight calculating method TF*IDF [10] to calculate BE score. Let D be the source document set for summarization, d_k denotes the k^{th} document in D, s_{jk} be the j^{th} sentence in document d_k, BE_{ijk} be the i^{th} BE in sentence s_{jk}, H_{ijk} be the head-BE of BE_{ijk}. The score of BE_{ijk} is defined as follows:

$$S'_{BE}(BE_{ijk}) = -\log(1+TF(H_{ijk}))*\log(IDF(H_{ijk})) \ . \tag{1}$$

Where TF(H_{ijk}) denotes the number of occurrence of H_{ijk} in document d_k, $IDF(H_{ijk}) = \log\frac{\#\text{documnets-contain-}H_{ijk}}{\#\text{documents}}$ is also known as "Inverted Document Frequency" which is computed over the documents in a large corpus (we use BNC corpus in this work).

Finally, the score is normalized among the documents:

$$S_{BE}(BE_{ijk}) = S'_{BE}(BE_{ijk}) / \max_{\hat{i}\,\hat{j}}(S'_{BE}(BE_{\hat{i}\hat{j}k})) \ . \tag{2}$$

3.1.3 Sentence Score Calculation

The score of a sentence is the summation of two weighted scores: the average score of its BEs and the score of the sentence position. Because the sentence occurs in the beginning of the document is more important, sentence position feature should be taken into account when calculating the sentence score. Suppose sentence s_{jk} contains l_{jk} BEs, document d_k contains n_k sentences. The score of sentence s_{jk} is calculated by formula (3):

$$S_S(s_{jk}) = \frac{\alpha}{l_{jk}}\sum_{i=1}^{l_{jk}} S_{BE}(BE_{ijk}) + (1-\alpha)\frac{n_k - j + 1}{n_k} \ . \tag{3}$$

Where α is the weight of BE score, $(1-\alpha)$ is the weight of position score. We let α =0.8 in this work.

3.2 BE Clustering

To process sentences in different documents as a whole, we create a sentence list SL that contains all of sentences in the document set D.

3.2.1 Sentence Representation

Vector space model (VSM) [11] handle massive real documents by adopting the existing mathematical instruments. In this paper, the BEs extracted from all the documents are used to represent the feature vector in VSM. In order to reduce the influence from BEs of little importance, those BEs with score less than half of average BEs score are removed. According to this, we set up the sentence VSM, where each sentence s_i in SL is represented as the weights of BEs, VSi. VS_i= (WB_{i1},WB_{i2},…,WB_{iN}), i=1,2,…M. where $M = \sum_{d_k \in D} n_k$ is the number of sentences in SL, N is the total number of remained BEs in document set D, WB_{ij} denotes the weight of the j^{th} BE in the i^{th} sentence. In this paper, we adopt TF*IDF to calculate WB_{ij} :

$$\mathrm{WB}'_{ij} = \text{-}\log(1+\mathrm{TF}(\mathrm{BE}_{ij}))*\log(\frac{M_j}{M}) \ . \tag{4}$$

Where $\mathrm{TF}(\mathrm{BE}_{ij})$ denotes the number of occurrence of the j^{th} BE in the i^{th} sentence, M_j/M denotes the inverted sentence frequency of BE_{ij}, and M_j denotes the number of sentence in which BE_{ij} occurs.

Finally, WB_{ij} is normalized as follows:

$$\mathrm{WB}_{ij} = \mathrm{WB}'_{ij} / \max_{\hat{i}\hat{j}} (\mathrm{WB}'_{\hat{i}\hat{j}}) \ . \tag{5}$$

3.2.2 K-Means Clustering

The k-means clustering method [12] is a fine choice in many circumstances due to its effectiveness with the complexity of O(nkt), where n is the number of sample points, k is the number of clusters and t is the number of iteration. We regard each sentence as a sample point in the N-dimensional sample space, and the sample space contains M sample points, where N is the number of all BEs in the document set D and M is the number of sentences.

To use the k-means method, the distance between two sentences must be defined. The calculation of sentence distance can be achieved by calculating the BE-vector distance. Generally, the cosine method is employed to calculate the similarity between two BE-vectors.

$$\begin{aligned}\mathrm{SIM}(\mathrm{VS}_i, \mathrm{VS}_j) &= \cos(\mathrm{VS}_i, \mathrm{VS}_j) \\ &= \frac{\sum_{t=1}^{N} \mathrm{WB}_{it} \bullet \mathrm{WB}_{jt}}{\sqrt{\sum_{t=1}^{N} \mathrm{WB}_{it}^2}\sqrt{\sum_{t=1}^{N} \mathrm{WB}_{jt}^2}} \ .\end{aligned} \tag{6}$$

Correspondingly, the distance between two BE-vectors can be calculated by the following formula:

$$\mathrm{DIS}(\mathrm{VS}_i, \mathrm{VS}_j) = 1 - \mathrm{SIM}(\mathrm{VS}_i, \mathrm{VS}_j) \ . \tag{7}$$

Figure 2 presents the formal description of the BE-vector clustering process based on the k-means method.

Input: the BE-vectors and the cluster number *K* (2 to *M*-1).

Output: *K* clusters

1) randomly select *K* BE-vectors as the initial centres of the clusters;

2) repeat:

– assign each BE-vector to the nearest cluster according to its distance to the cluster centres;

– recalculate the new centre for each cluster;

3) until the change of centres is very little.

Fig. 2. BE-vector clustering process using k-mean method

3.2.3 Automatic Determination of *K*

A classical problem with the k-means clustering method and many other clustering methods is the determination of K, the number of clusters. In the traditional k-means method, K must be decided by the user in advance. In many cases, it's impractical. As for BE clustering, user can't predict the latent cluster number, so it's impossible to offer K correctly.

In this paper, two kinds of methods are proposed to detect *K* automatically.

The first method is simple and inspired by the limited summary length fixed by the user. On the one hand, summary length is usually fixed by user, so the number of extracted sentences is approximatively fixed at the same time. On the other hand, to generate an anti-redundant summary, summarizer usually extracts only one sentence from each cluster. So, the number of sentences in fixed-length-summary is an acceptable value for the number of clusters. The most probable number of sentences in a fixed-length-summary is the length of summary divided by the average length of sentences in document set. Thus, we determine the approximate number of clusters as:

$$K' = L_{\mathrm{SM}}/\mathrm{avg}(L_{\mathrm{S}}) \ . \tag{8}$$

Where L_{SM} denotes the summary length fixed by the user, $\mathrm{avg}(L_{\mathrm{S}})$ denotes the average length of sentences in the document set *D*.

The basic idea of the second strategy is that if the cluster number *K* is correct, the within-cluster-similarity of vectors should be higher whereas the between-cluster-similarity of vectors should be lower.

We define the cohesion of a cluster and scatter between two clusters as formula (9) and (10) respectively.

$$\mathrm{CHN}(c_i)=\frac{2}{|c_i|(|c_i|-1)}\sum_{\substack{\mathrm{VS}_p,\mathrm{VS}_q\in c_i\\ \mathrm{VS}_p\neq \mathrm{VS}_q}}\mathrm{SIM}(\mathrm{VS}_p,\mathrm{VS}_q) \ . \tag{9}$$

$$\mathrm{SCT}(c_i,c_j)=\frac{1}{|c_i||c_j|}\sum_{\mathrm{VS}_q\in c_i}\sum_{\mathrm{VS}_p\in c_j}\mathrm{DIS}(\mathrm{VS}_p,\mathrm{VS}_q) \ . \tag{10}$$

Where c_i is the i^{th} cluster generated by the k-means clustering method. $|c_i|$ is the number of elements (members) in cluster c_i.

The evaluation function of the clustering result is defined as follows:

$$\mathrm{F}(C)=\frac{1}{K}\sum_{c_i\in C}\mathrm{CHN}(c_i)+\frac{2}{K(K-1)}\sum_{\substack{c_i,c_j\in C\\ c_i\neq c_j}}\mathrm{DSP}(c_i,c_j) \ . \tag{11}$$

Where *C* is the cluster set of clusters, which is the result of the k-means method.

The number of clusters is determined by maximizing the evaluation function F(C):

$$K^{*} = \underset{K\in\{2,\ldots,M-1\}}{\operatorname{argmax}}\ \mathrm{F}(C)\ . \tag{12}$$

3.3 Sentence Selection

The easiest way to select sentences from the sentence list is to output the topmost sentence from each cluster until the required summary length limitation is reached. However, this simple approach does not consider the relation between length of summary and number of clusters. Suppose we get K clusters after the k-means algorithm presented above, the total length of K topmost sentences from each cluster may be longer or shorter than the required summary length limitation. In this paper, we re-sort the sentence list in descendant order according to the sentence score at first, and then select sentences from the clusters repeatedly according to the sentence order in sentence list. Figure 3 presents the detail process of this method.

```
Input: sentence list (attributes of element: sentence no., sentence score, sentence length and cluster no. this sentence is assigned to), the required summary length $L_{\mathrm{SM}}$.
Output: a set of the selected sentences.
( Let $M$ be the number of elements in sentence list, $s_i$ be the $i^{th}$ element in sentence list, SN($s_i$) be the sentence no. of $s_i$, CN($s_i$) be the cluster no. that $s_i$ is assigned to, $c_i$ be the $i^{th}$ cluster in cluster set $C$, HBS($c_i$) be the number of sentences have been selected from cluster $c_i$, LEN($s_i$) be the length of $s_i$, $L_{\mathrm{SM}}$ be the required summary length, SLC be the set of selected sentences.)
1) Resort the sentence list SL in descendant order according to the sentence score.
2) For $i$ from 1 to $M$
3) If $s_i$ satisfies the following two conditions
       a. $\mathrm{HBS}(c_{\mathrm{CN}(s_i)}) \le \min_{c_k \in C}(\mathrm{HBS}(c_k))$ ;
       b. $\mathrm{LEN}(s_i) + \sum_{s_j \in \mathrm{SLC}} \mathrm{LEN}(s_j) \le L_{\mathrm{SM}}$ ;
   then
       add $s_i$ in SLC;
       recalculate $\mathrm{HBS}(c_{\mathrm{CN}(s_i)})$
4) output SLC
```

Fig. 3. The sentence selection method

3.4 Summary Generation

Finally, the selected sentences are output according to their positions in the original document to form the final summary. To improve the consistency of the final summary, the original document set should be sorted by the temporal order.

4 Experimentation

4.1 Experimental Setting

The data used in this work is the document set for task 1&2 in DUC04 [13]. There are 50 sets of English TDT documents. Each set contains 10 documents. Task 2 of DUC04 requires participants produce a short summary no more than 665 bytes for each document cluster. Four human model summaries are provided for each cluster for evaluation.

ROUGE [14] stands for recall-oriented understudy for gisting evaluation. It includes measures to automatically determine the quality of a summary by comparing it with ideal summaries created by humans. The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries created by humans. DUC04 use ROUGE-1,2,3,4, ROUGE-L, and ROUGE-W to measure summaries generated by participants. We follow the same requirement of DUC04 task 2. All ROUGE evaluation configurations also follow the configurations used in DUC04 by using the same command and options: stop words included, porter stemmed and use only the first 665 bytes.

4.2 Evaluation

Figure 4 illustrates the results of two methods for K detection on 50 document sets. K' and K^* are the numbers of clusters detected using formula (8) and (12) respectively. We can shrink the search space of formula (12) from [2,M-1] to [2, 2K'] on two reasons: one is that the K^* detected by formula (12) has not much great discrepancy compared with K', the other is that the required summary length limits the number of clusters.

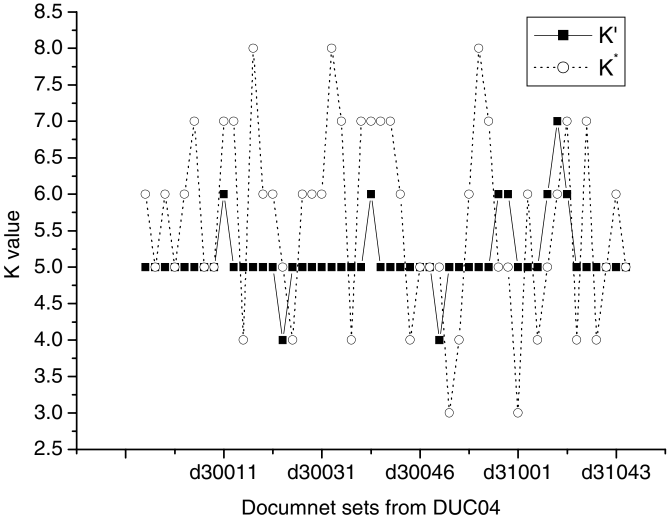

Fig. 4. Results of K detection method (K' using formula (8); K^* using formula (12))

Table 1 lists the ROUGE scores of summaries using MSBEC and summaries using the word-vector clustering strategy (MSWC for abbreviation). To compare our strategy with DUC04 participants, this work re-evaluated all summaries generated by participants using ROUGE1.5.5 package (note that the results are of neglectable difference between ROUGE1.5.5 package and ROUGE package in DUC04). Table 1 lists the average scores of human summaries and the scores of best peers generated by participants as well (unfortunately, there is no paper submission for the best system in DUC 04). Evaluation results show that the BE-vector clustering strategy (MSBEC) is superior to the word-vector clustering strategy (MSWC) for multi-document summarization. The comparison between MSBEC and the best system on DUC04 demonstrates that our strategy is effective.

Table 1. Rouge score comparison

N-gram (F-measure)	Average Human Peers	Best System	MSBEC	MSWC	MSBEC VS. MSWC	MSBEC VS. Best system
Rouge 1	0.40441	0.37917	0.37291	0.36936	+0.96%	-1.65%
Rouge 2	0.09665	0.09152	0.08951	0.08570	+4.44%	-2.20%
Rouge 3	0.03021	0.03332	0.03214	0.03017	+6.53%	-3.54%
Rouge 4	0.01094	0.01533	0.01433	0.01353	+5.86%	-6.56%
Rouge L	0.36193	0.32757	0.32371	0.32194	+0.548%	-1.18%
Rouge w1.2	0.15897	0.14691	0.14499	0.14408	+0.63%	-1.31%

5 Conclusions

In this paper, we have proposed a new multi-document summarization strategy based on BE-vector clustering. Because BEs can represent high-informative unigrams, bigrams, and longer units of a text, the performance of multi-document summarizer can be improved by using BE as the minimal semantic unit. Experiments on DUC04 data set proved the efficiency of our strategy. Moreover, we adopted a novel clustering analysis method to automatically detect the number of clusters in the k-means clustering method. For the future work, we will explore more features and apply the BE-vector clustering strategy in query-based multi-document summarization system.

References

1. Radev Dragomir, Jing Hongyan, Budzikowska Malgorzata: Centroid-Based Summarization of Multiple Documents: Sentence Extraction, Utility-Based Evaluation and User Studies. Information Processing and Management, Vol. 40. (2004) 919–938
2. Hardy Hilda: Cross-Document Summarization by Concept Classification. In Proceedings of the 25th annual international ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press (2002) 121–128
3. Mitra M., Singhal Amit, Buckley Chris: Automatic Text Summarization by Paragraph Extraction. In ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain (1997) 31–36

4. Knight Kevin, Marcu Daniel: Summarization Beyond Sentence Extraction: a Probabilistic Approach to Sentence Compression. Artificial Intelligence, Vol. 139. (2002) 91–107
5. Barzilay Regina, McKeown Kathleen R., Michael Elhadad: Information Fusion in the Context of Multi-Document Summarization. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, New Jersey: Association for Computational Linguistics (1999) 550–557
6. Manuel J, MAN`A-LO`PEZ: Multi-document Summarization: An Added Value to Clustering in Interactive Retrieval, New York: ACM Transactions on Information Systems, Vol. 22. (2004) 215–241
7. Po Hu, Tingting He, Donghong Ji, Meng Wang: A Study of Chinese Text Summarization Using Adaptive Clustering of Paragraphs. In Proceeding of the Fourth International Conference on Computer and Information Technology (CIT'04), Wuhan, (2004) 1159-1164.
8. Eduard Hovy, Chin-Yew Lin, Liang Zhou, Junichi Fukumoto: Basic Elements. Technical Report, http://www.isi.edu/˜cyl/BE/index.html (2005)
9. Dekang Lin: Minipar. http://www.cs.ualberta.ca/˜lindek/minipar.htm (1998)
10. Baeza Yates R., Ribeiro Neto B.: Modern Information Retrieval. New York: Addison Wesley (1999) 27–30
11. Patrick Pantel, Dekang Lin: Document Clustering with Committees. In Proceedings of ACM, SIGIR'02, New York: ACM (2002) 199–206
12. Addrew R. Webb: Statistical Pattern Recognition, 2nd edn. John Wiley & Sons (2002) 376–379
13. Over Paul, Yen James: An Introduction to DUC-2004. In Proceedings of the 4th Document Understanding Conference (DUC 2004) (2004)
14. Chin-Yew Lin, Eduard Hovy: Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. In Proceedings of the Human Technology Conference (HLTNAACL-2003), Edmonton, Canada (2003)