A BE-based Multi-document Summarizer with Sentence Compression

Eduard Hovy, Chin-Yew Lin and Liang Zhou
Information Sciences Institute
University of Southern California
Marina del Rey, CA90292
USA
{hovy,cyl,liangz}@isi.edu

Abstract

This paper describes a multi-document summarizer based on basic elements (BE), head-modifier-relation representation of document content developed at ISI. To increase the coverage of automatically created summaries at a given length, we first generate a summary about twice of the intended length, then apply compression techniques to make sure the resulting summaries fall within the length constraint of target summaries. Our initial results show that the BE-based summarizer with compression achieved 0.0654 in BE-F score that was significantly better than the BE-F score of 0.0542 without compression.

1 Introduction

This paper describes a multi-document summarizer based on basic elements (BE) (Hovy et al. 2005), a head-modifier-relation triple representation of document content developed at ISI. BEs are intended to represent the high-informative unigrams, bigrams, and longer units of a text, which can be built up compositionally. An important aspect is that they can be produced automatically. However, BEs can also be used as a counting unit for frequency-based topic identification. The idea is to assign scores to BEs according to some algorithms, assign scores to sentences based on the scores of the BEs contained in the sentences, and then apply standard filtering and redundancy removal techniques before generating summaries. To increase the coverage of automatic summaries at a given length, we first generate a summary about double of the intended length, then apply compression techniques to make sure the resulting summaries fall under the length constraint of target summaries. Our experimental results show that this approach was very effective in MSE 2005. We give a short overview of Basic Elements in the next section. Section 3 describes the BE-based multi-document summarizer. Section 4 presents our sentence compression method and we conclude and discuss future directions in Section 5.

2 Basic Elements

At the most basic level, Basic Elements are defined as follows:

- the head of a major syntactic constituent (noun, verb, adjective or adverbial phrases), expressed as a single item, or
- a relation between a head-BE and a single dependent, expressed as a triple (head | modifier | relation).

BEs can be created automatically in several ways. Most of them involve a syntactic parser to produce a parse tree and a set of 'cutting rules' to extract just the valid BEs from the tree.

With BE represented as a head-modifier-relation triple, one can quite easily decide whether any two units match (express the same meaning) or not–considerably more easily than with longer units, of the kind that have been suggested for summarization evaluation by other researchers (Van Halteren and Teufel, 2003; Nenkova and Passonneau, 2004). 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.

Example BEs for "two Libyans were indicted for the Lockerbie bombing in 1991" are as follows, written as (head | modifier | relation):

libyans|two|nn (BE-F)
indicted|libyans|obj (BE-F)
bombing|lockerbie|nn (BE-F)
indicted|bombing|for (BE-F)
bombing|1991|in (BE-F)

The BEs shown above (BE-Fs) are generated by BE package 1.0 distributed by ISI[1]. We used the standard BE-F breaker included in the BE package in all our experiments described in this paper.

3 BE-based Multi-document Summarizer

We modeled our BE-based multi-document summarizer after the very successful NeATS (Lin and Hovy 2002). It includes the following three major stages.

(1) Identify Important BEs

BEs were used as counting unit. We replaced unigram, bigram, and trigram counting in NeATS with BE-F counting, i.e. breaking each sentence into BEs instead of unigrams, bigrams, and trigrams. We then computed likelihood ratio (LR) for each BE. The LR score of each BE is an information theoretic measure (Dunning, 1993; Lin and Hovy, 2000) that represents the relative importance in the BE list from the document set that contains all the texts to be summarized. Sorting BEs according to their LR scores produced a BE rank list.

(2) Identify Important Sentences

The score of a sentence is the sum of its BE scores computed in (1) divided by the number of BEs in the sentences. We call this normalized sentence BE score. Sorting sentences according to their normalized sentence BE scores produced a ranked list of sentences. By limiting the number of top BEs that contribute to the calculation of sentence scores, we can remove BEs with little importance and sentences with many less important BEs. We call this parameter B. For example, B = 64 means that only the topmost 64 BEs in the rank list created in (1) can contribute to normalized sentence BE score computation.

(3) Generate Summaries

The easiest way to create summaries from (2) is just to output the topmost N sentences until the required summary length limit. However, this simple approach does not consider interactions among summary sentences, such as redundancy and coherence. For example, we should only include one of two very similar sentences with high normalized sentence BE scores in a summary. Goldstein et al. (1999) observed this in what they called maximum marginal relevancy (MMR). This we modeled by BE overlap between an intermediate summary and a to-be-added candidate summary sentence. We call this overlap ratio R, where R is between 0 and 1 inclusively. For example, R = 0.8 means that a candidate summary sentence, *s*, can be added to an intermediate summary, *SI*, if the sentence has a BE overlap ratio less than or equal to 0.8.

Also, given the importance in the news genre of sentence position (Lin and Hovy, 1997), we would like to model the position preference that favors sentences appearing earlier in a document. This is controlled by parameter N. For example, N = 10 means that only the first 10 sentences in a document can be considered as candidate summary sentences.

In favor of leading sentences of the news genre and provide a simple way to improve coherence (lead sentences usually give the setting of news events), we adopted a first-sentence-priority policy, i.e. if a to-be-added candidate summary sentence is not a lead sentence and its lead sentence[2] is yet not included in the immediate summary, then add its lead sentence first when its addition does not violate the overlap ratio constraint. This strategy was used with considerable success in NeATS.

Through experimentation using the DUC 2003 task 2 corpus, we found that the BE-based multi-document summarizer with B = 64, R = 0.8, and N = 10 achieved a BE-F score of 0.0532 that was better than the summaries generated by NeATS (at 0.0503) in DUC 2003. We therefore decided to use this set of parameters in MSE 2005[3].

4 Sentence Compression

The discussion so far has focused on extraction. However, identifying salient information is only the first half of the summarization problem. A number of researchers have started to address the

[1] BE website: http://www.isi.edu/~cyl/BE

[2] The lead sentence of a document is the lead sentence of all the sentences in the document.

[3] Note that we did not have time to finish the three-stage summarization procedure described here before the submission deadline of MSE 2005. All the numbers reported here were post-deadline internal experiments and evaluations.

possibility of generating coherent summaries through summary revisions (Mani et al., 1999) and regeneration from derivation of information/theme intersection (Barzilay et al., 1999). In particular, Knight and Marcu (2000) have proposed to find the balance between grammaticality and textual retainment through both a noisy-channel model and a decision-based model. Both models learn from a training corpus to perform tree-reduction operations either probabilistically or through example learning. These two models mainly focus on syntactic tree learning and performing syntactic-based compressions, rather than information preservation.

The extraction mechanism of our summarization system selects sentences that share a certain degree of information overlap. The overlap acts as a bridging medium for extracting sentences that, when selected together, would produce a higher volume of important textual content. Valuable information is identified by top-ranked BEs, indicating a high occurrence of repetition. Through experimentation, we discovered that BEs appearing in the same context (in this case, in the same sentence) also carry some degree of textual importance. This is shown with the increase in recall scores when those secondary BEs are included in the summary results. Therefore, ideally we would like to remove redundant information while producing the most coherent sentence sets. While this goal is shared by all compression research for summarization, it is yet to be realized while performing actual compression operations. Either syntactic structure or information content is taken into account primarily, but not both at the same time.

We envisage a compression technique where reduction operations are performed on parse trees' syntactic constituents marked for "removal". In addition, the compression module will decide the most appropriate level of the tree the marked constituents shall be cut from.

4.1 Content Labeling

The compression procedure is invoked incrementally. Sentences selected by the extraction module are ranked according to the importance of their information content, i.e. the total weight of top-ranked BEs normalized by sentence length. A list of top-ranked BEs is then maintained for each document set. Each sentence is presented by its BE equivalent. For example, the sentence "A man was killed by police." becomes:

killed | man | obj
killed | by | by-subj
killed | police | by .

The first sentence from the extract contains the most salient information from the document set, with no compression applied. Any sentence following it should only complement its content with additional information.

Top-ranked BEs from the first sentence are recorded in a "have-seen" table. Before a second sentence is added, all of its BEs are checked against the "have-seen" table. If any of the BEs appear in the table, they are labeled as "remove". Top-ranked BEs from this sentence are then also recorded in the same "have-seen" table.

This procedure is performed on every sentence from the extract. The "have-seen" table is maintained globally and the "remove" lists are maintained on per sentence basis.

4.2 Parse Tree Reduction

Knight and Marcu use sentence pairs of the form (*long sentence, abstract sentence*) from the Ziff-Davis corpus (newspapers with abstracts) to collect *expansion-template* probabilities. Expansion templates are created through identifying corresponding syntactic nodes (Collins, 1997) from those sentence pairs. Assigning probabilities to trees rather than strings would introduce an information loss. For summarization tasks with a strict length limit we should constrain this kind of loss to a minimum. The challenge is to perform tree reduction with information retention as a priority.

BEs are minimal semantic units. If we could compress sentences by identifying the smallest yet necessary removable units and remove them correctly according to grammar rules, then minimum information loss and maximal grammaticality can be achieved.

As stated in the previous section, BEs for sentences have been labeled as "remove" or "keep." A parse tree is also produced for each sentence using Collins' parser. In Figure 1, we show part of a parse tree where the compression would take effect. The smallest (furthest down the tree) constituent that covers a "remove" BE is first identified. Its

ancestors (parent, grandparent, etc.) are traversed, and at each ancestor node, we assume the larger tree can be cut. (Figure 1, edges labeled "1" and "2"). For each resulting smaller tree t, $P_{tree}(t)$ is computed over the Penn TreeBank PCFG grammar rules that yielded the tree t. Among the smaller trees, the one that has the highest $P_{tree}(t)$, normalized by the number of grammar rules used, is considered as the best candidate tree. But if one of its children (not containing the "remove" BE) contains unseen top-ranked BEs, the tree-cutting operation that produced this tree should not be activated. In other words, if the cutting operation at an ancestor level was deemed not desirable because one of the ancestor's children contains important non-redundant information, the compression module backtracks to the next level down in the tree. This process is performed at every node, traversing from the lowest tree level that covers the "remove" BE up until a decision to backtrack is made on one of the upper-level ancestors.

From the sample shown in Figure 1, let's assume that the BE "diplomatic immunity" has appeared in a previous sentence and needs to be removed. From computing $P_{tree}(t)$ for t_1 and t_2 (edges labeled as "1" and "2" in tree), let's assume t_1 is preferred. But if somehow the BE "be entitled (is entitled)" is one of the top-ranking BEs and needs to be kept in the sentence, then t_2 is the preferred tree.

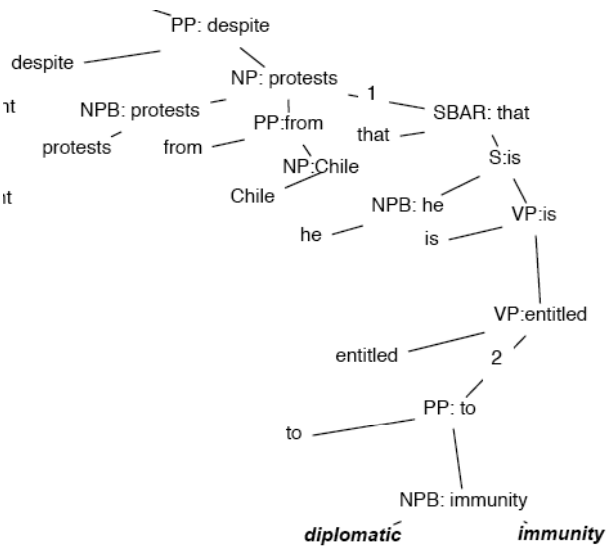

Figure 1. An example for sentence compression.
Original sentence:
Former Chilean dictator Gen. Augusto Pinochet has been arrested by British police on a Spanish extradition warrant despite protests from Chile that he is entitled to diplomatic immunity.
Removable BE(s):
diplomatic immunity
t1:
Former Chilean dictator Gen. Augusto Pinochet has been arrested by British police on a Spanish extradition warrant despite protests from Chile.
t2:
Former Chilean dictator Gen. Augusto Pinochet has been arrested by British police on a Spanish extradition warrant despite protests from Chile that he is entitled.

4.3 Validation

The compression mechanism is designed for summarization. Therefore, summarization evaluation methodologies should be used to evaluate the now-compressed summaries. The newly introduced and publicly available Basic Element evaluation tool kit is used in our experiment.

At 100 words, the best multi-doc uncompressed extracts generated from DUC2003 data result in a recall of 0.0532 on BE-F. 200-word extracts, before compression, chart a 0.0786 in BE-F recall. When compression is applied, resulting in 100-word summaries, we see a significant improvement in BE-F recall at 0.0578.

The preliminary results are quite encouraging. The compression would be much more effective if such a corpus for training were available. The decision process on each tree node would be probabilistic, rather than ad hoc.

5 Conclusions

We have shown that a BE-based multi-document summarizer with sentence compression can achieve significant improvement over non-compressed extraction-only summarizer. However, the loss of information, from 0.0786 to 0.0578 in DUC 2003 and 0.09413 to 0.0654 in MSE 2005 due to the compression procedure (from 200 words to 100 words) described in Section 4 still leave much room for improvement. We plan to create a summary compression corpus and train a probabilistic summary compressor to replace the current rule-based approach. With the encouraging

results in the post-evaluation experiments, we are confident that BE-based multi-document summarization with probabilistic sentence compression is an interesting and very promising research direction.

References

Barzilay, R., McKeown, K., and Elhadad, M. 1999. Information Fusion in the Context of Multi-Document Summarization. *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, 1999, Maryland.

Collins, M. 1997. Three generative, lexicalized models for statistical parsing. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97),* 16-23.

Dunning, T. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1), 61-74.

Goldstein, J., M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. *Proceedings of the 22nd International ACM Conference on Research and Development in Information Retrieval* (SIGIR-99), Berkeley, CA, 121-128.

Hovy, E. H., J. Fukumoto, C.-Y. Lin, L. Zhou. 2005. Basic Elements. http://www.isi.edu/~cyl/BE

Knight, K., and Marcu, D. 2000. Statistics-Based Summarization: Step One: Sentence Compression. *Proceedings of the 17th National Conference of the American Association for Artificial Intelligence AAAI 2000*, Austin, Texas, July 30-Augeust 3, 2000.

Lin, C-Y. and E. Hovy. 1997. Identify Topics by Position. *Proceedings of the 5th Conference on Applied Natural Language Processing* (ANLP), Washington, D.C.

Lin, C-Y, and E. H. Hovy. 2000. The Automtated Acquisition of Topic Signatures for Text Summarization. *Proceedings of the COLING Conference*, Strasbourg, France. August 2000.

Lin, C.-Y. and E. Hovy. 2002. Automated Multi-document summarization in NeATS. *Proceedings of the Human Language Technology Conference* (HLT2002), San Diego, CA, USA, March 23-27, 2002.

Mani, I., Gates, B., and Bloedorn, E. 1999. Improving Summaries by Revising Them. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL '99)*, 558-565.

Nenkova, A. and R. Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. *Proceedings of the HLT-NAACL conference*. Boston, MA.

Van Halteren, H. and S. Teufel. 2003. Examining the Consensus between Human Summaries: Initial Experiments with Factoid Analysis. *Proceedings of the HLT-NAACL Workshop on Automatic Summarization*. Edmonton, Canada.