The TITech Summarization System at TAC-2009

Yuanrong Zheng, Tokunaga Takenobu
Department of Computer Science
Tokyo Institute of Technology
{zhengyr, take}@cl.cs.titech.ac.jp

Abstract

This paper presents the TITech summarization system participating in TAC2009. Specifically, we discuss our results for the *Update* track. We propose a new method for creating summaries by ordering sentences. After a draft summary is obtained, we conduct agglomerative hierarchical clustering on the sentences of the draft summary based on sentence associativity. Then we use a probabilistic method to adjust the order of these draft summaries. We submitted two runs: the results that sentence order is decided by utilizing chronological information, and the results of our proposed method.

1 Introduction

The update summarization task in TAC 2009 is to produce a 100-word summary for each batch of articles under the assumption that the user has already read a set of earlier, related articles. The update summary of the second batch of articles should inform the user of new information about the topic. The documents for summarization come from the AQUAINT-2 collection of news articles.

We propose a new method that reorders sentences for creating more readable summaries. In our experiments, we try to investigate the effects of sentence ordering and its influence on the quality of the resultant summary.

First, we generate a draft summary for each document set. Next, we use different methods to rearrange the sentence order of the draft summary. Depending on the summary length requirement, we may then truncate the overall summary. We submitted two runs: the first contains the results for when sentence order is decided by utilizing chronological information, and the second contains the results of our proposed method.

The rest of this paper is organized as follows. We first give an overview of our system in detail, specifically we address the sentence ordering module of our system. We then report evaluation results from NIST. In Section 4, we discuss future work and conclude this paper.

2 System Description

2.1 System Structure

Our system consists of three main components, typical for summarization systems: (1) document preprocessing, (2) summary generation and (3) postprocessing.

2.1.1 Preprocessing

The preprocessing step includes sentence segmentation, POS tagging, word stemming, and ordering model creation (to prepare for the last step of summarizing) as outlined below. We use the Python Natural Language Toolkit (Bird and Loper 2004) in the preprocessing part in our system. NLTK is written in Python and distributed under the GPL open source license.

Sentence Segmentation The sentence is often considered the basic element in extractive summarization. We extract the content from the documents and segment them into sentences.

POS tagging Nouns and verbs are content words; they can be thought of as having a strong connection with the topic of the document in which they appear. We consider only nouns and verbs in our system, so POS tagging is necessary.

Word Stemming In English, many words in different forms but with the same root share the same meaning, e.g. clued and clue. We hypothesize that sentence similarity is better measured not by using the words as they appear, but by using their roots.

Ordering Model Creation Sentence ordering is important for multi-document summarization, especially as the summarized content gets longer in length. The words' physical position can be used as a hint to help decide how to best arrange the sentences in the summary. For each source document, we collect the words' physical position information.

Given two segments S_i and S_j, their order is defined below.

$$O_{ij} = \sum_{k,l} O_f\ (f_{ik}, f_{jl}). \tag{1}$$

Here f_{ik} represents the features of segment S_i, and f_{jl} the features of segment S_j. O_f is the ordering function of a feature pair (f_i, f_j). $O_f(f_{ik}, f_{jl})$ is the ordering weight of features f_{ik} and f_{jl} in all source documents. If O_{ij} is positive, segment S_i should come before S_j, otherwise, segment S_i should come after S_j.

The ordering weight of feature pair f_i, f_j is defined below.

$$O_f\ (f_i, f_j) = \sum_d (F\ (f_i, f_j) - F\ (f_j, f_i)). \quad (2)$$

Here F is the frequency function, and $F\,(f_i, f_j)$ denotes the frequency of feature f_i appearing before feature f_j in the source documents. The same holds true for $F\,(f_j, f_i)$, though it describes the frequency of f_j appearing before f_i.

Since scanning the entire document before the occurrence of any feature is computationally expensive we limit the frequencies empirically to be within 4 sentences. The features currently processed are nouns and verbs, stemmed as mentioned above.

In this step, we calculate all possible $F\,(f_i, f_j)$.

2.1.2 Summary Generation

To evaluate whether it is appropriate to include a given sentence within the summary or not, several sentential features are considered:

Sentence Location: The position of sentences in a document can play a significant factor in finding the sentences that are most related to the topic of the document. We therefore take into account the position of each sentence when computing its score. We give the score $1/n$ to the n-*th* sentence in each paragraph.

Named Entities: Using NLTK, it is possible to recognize the Named Entities (NEs) mentioned in each document. The sentences containing more NEs are assumed to be more important than those that contain less. Note that only the frequency of NEs in each sentence is taken into account when forming the scoring formula.

Topic Statement: The topic statement, provided in the test data for each of 44 clusters, characterizes the set of documents and is without doubt of paramount importance to quantify the relevance of each sentence with respect to the overall meaning that is conveyed by the documents. Therefore, the evaluated semantic similarity of each sentence and topic statement is explicitly taken into account.

Sentence Importance: Given a sentence collection $S = \{s_i | 1 \leq i \leq n\}$, the similarity $sim(s_i, s_j)$ between a sentence pair s_i and s_j is calculated by using the cosine measure. The weight associated to term t is calculated by applying the $tf_t * isf_t$ formula, where tf_t is the frequency of term t in the corresponding sentence and isf_t is the inverse sentence frequency of term t, i.e. $1 + log(N/n_t)$, where N is the total number of sentences and n_t is the number of sentences containing term t. To calculate the sentence similarity, we only consider nouns and verbs; in addition, we use WordNet to unify words with synonymous meanings. After computing sentence similarity, we use an affinity graph based method (Wan, 2006) to decide sentence importance.

Update information: The idea behind an update summary is that it should predominantly contain new information, since the reader is already familiar with previously read documents on the subject. Yet, it is hard to define which information is new. However, we can use set A (the already read document set) to find information that is *not* new. We can then punish the words that appear in both set A and set B (the new, unread document set) to find any new information by diminishing their weights. Then we sum up all the weights that a sentence contains as its update information score.

The score for each sentence is generated based on the linear combination of the weighted features computed in the previous steps. The formula used for scoring each sentence is:

$$\begin{aligned} Score(i) = \quad & \alpha SL(s_i) + \beta NE(s_i) + \gamma T(s_i) + \\ & \delta SI(s_i) + \epsilon U(s_i), \end{aligned}$$

where

- $SL(s_i)$ is the sentence location weight,
- $NE(s_i)$ is the number of named entities in the document,
- $T(s_i)$ is the semantic similarity between the sentence and the topic statement,
- $SI(s_i)$ is the score for sentence importance,
- $U(s_i)$ is the update information score.

Before combination, each factor should be normalized, divided by the largest score, and the weighting parameters α, β, γ, δ, ϵ ($\alpha + \beta + \gamma + \delta + \epsilon = 1$) are user chosen, depending on his/her prior knowledge about the relevance of these features. In the absence of any further evidence, the default value for each is experimentally set to 0.2.

With this ranked list of all the sentences we can make a draft summary for each document set consisting of the highest scored sentences from the list. We then proceed to use the sentence ordering method introduced in the next section to rearrange the sentences for generating the final summary.

2.1.3 Postprocessing

After we get the ranked list of scores for all the sentences, redundancy removal is performed, resulting in a draft summary. Maximal Marginal Relevance(MMR) (Carbonell and Goldstein, 1998; Goldstein et al., 2000) balances relevance and anti-redundancy by selecting one sentence at a time for inclusion in the summary and re-scoring for redundancy after each selection. After redundancy removal, we adjust sentence order to create a more readable summary. We will introduce this in detail in the next section.

2.2 Sentence Ordering

2.2.1 Related Work

The issue of sentence ordering is important for natural language tasks such as multi-document summarization, yet the area still lacks substantial exploration concerning the range of acceptable sentence orderings for short texts. For Multi-document summarization (MDS) systems, it is important to determine a coherent arrangement of the textual segments extracted from the source documents in order to accurately reconstruct the text structure for summarization. A summary of the few methods that do exist for arranging sentences in MDS is provided below.

Chronological ordering (McKeown et al., 1999; Lin and Hovy, 2001; Barzilay et al., 2002; Okazaki et al.,2004) is often used for this task. It orders sentences by published date of source documents or time information within texts. Sometimes its results are good, but chronological information is not always available, not to mention sentences which have the same publication date fail to be ordered entirely.

The probabilistic model (Lapata, 2003) ordered sentences based on conditional probabilities of sentence pairs. The conditional probabilities of sentence pairs are learned from a training corpus. With conditional probability of each sentence pair, the optimal global ordering is estimated by a simple greedy algorithm.

Majority ordering (McKeown et al., 2001; Barzilay et al., 2002) groups sentences to be ordered into different subtopics and then arranges these subtopics according to the topic order in the source documents. However, how to best cluster sentences into topics, and how to best order sentences belonging to the same topic is not a trivial matter.

Adjacency ordering (Nie et al., 2006; Ji et al., 2008) uses the connectivity of each sentence pair: after the first sentence is decided, the next is the sentence with the highest connectivity with the previous one. In this serial structure, any unsuitable choice will affect all subsequent decisions.

The bottom-up method (Bollegala et al. 2006) combines chronological ordering, topical-closeness, precedence and succession together. With this method, however, one can not make sure that the adjacent sentences have a high degree of connectivity with each other.

2.2.2 Proposed Method

There are two heuristics that can be used in when ordering sentences, the first is that sentence pairs with high associativity should be adjacent to another another and the other is that the ordering of the sentences in the summary should be biased to be similar to how they appear in the source documents. In the method we propose, we try to utilize the two heuristics together. In our experiments, we investigate the effect of our method on sentence ordering and the influence of the sentence ordering on the quality of the summary.

First, we generate a draft summary for each document set. For a set of documents on a certain topic, there are usually several suitable summaries. So what is left is how to use some sentences from the draft summary to get a more readable final summary. We tried two methods to adjust the sentence ordering of the draft summary separately. As a last step, it is possible to truncate the summary according to a length requirement. One of the two runs we submitted is the results of our proposed method, and the other one is the results utilizing the chronological method.

We investigate a new way to order sentences in which the more associative sentences will be adjacent, but still allow for utilizing other information (such as chronological information and majority statistics). We consider the problem of modeling the content structure of documents on a certain topic in terms of the subtopics addressed by the documents and the order in which these subtopics appear. The model we use is a global model in which subtopic ordering is biased to be similar across a collection of related documents, while sentence pairs with high connectivity tend to be put together. Here, the associativity of each pair of sentences is learned from their source documents.

This is a two phase method. In the first phase, we conduct an agglomerative hierarchical clustering on the sentences of the draft summary based on sentence associativity. To cluster these sentences, we use the sentence similarity calculation method mentioned before. Here we produce a binary tree which confirms that the sentence pairs with high associativity tend to be put together. As shown in Figure 1, sentences 1 and 4 have the highest associativity, so we put them in one subtree ("segment"); this sub-tree will be seen as a node in next step. Next, among segment 1 and sentences 2 and 3, the sentence pair formed from sentences 2 and 3 has the highest associativity, so we put sentence 2 and 3 together. Last, we put segment 1 and 2 together, producing a tree in which the sub-trees can be seen as subtopics of the summary. The results of doing an inorder traversal on this tree provide a sentence order.

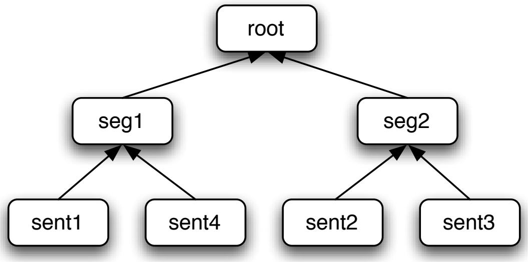

Figure 1: Content Structure Denoted by Binary Tree

The physical position of words from source documents can be used as a hint for arranging the sentences in the final summary. In the second phase, we adjust the adjacent segments to find a better sentence order. This is a top-down adjusting method. We examine both subtrees of each non-leaf node, where each feature pair consists of a word from each of the two sub-trees. If a sub-tree contains more words appearing before the words from the other sub-tree in the source documents, it is set as the left sub-tree. Utilizing the ordering model created before, for two sub-trees T_i and T_j, the sentences contained can be seen as a segment, i.e. S_i, S_j, if O_{ij} is positive, then T_i will be the left sub-tree and T_j will be the right. Otherwise, T_j will be the left subtree and T_i will be the right. As shown in Figure 2, the words in segment 1 mostly appear before words in segment 2 in the source documents, so we arrange segment 2 as the left sub tree and segment 1 as right.

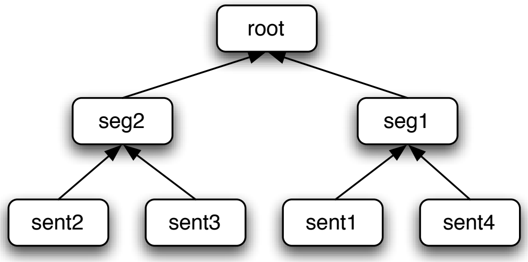

Figure 2: Binary Tree Adjusting: First Step

Next we consider the lower-levels of each sub-tree. Since the words in sentence 3 mostly appear before words in sentence 2, we put sentence 3 as the left node. For the same reason, we put sentence 4 as the left subtree. As shown in Figure 3. After adjustment, the sentences in the left sub-tree are always placed in the summary before the ones in the right sub-tree. The results of doing an in-order traversal on the tree should give a better sentence order.

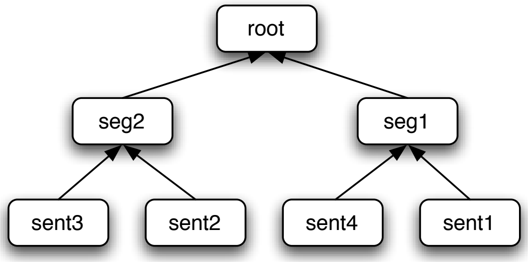

Figure 3: Binary Tree Adjusting: Second Step

2.3 Testing Corpus

The test data set for the TAC 2009 Update Summarization Task comprises 44 topics. Each topic has a topic statement (title and narrative) and 20 relevant documents which are divided into 2 sets: Document Set A and Document Set B. Each document set has 10 documents, where all the documents in Set A chronologically precede the documents in Set B. The document sets come from the AQUAINT-2 collection of news articles.

3 Evaluation and Discussion

There are 52 runs from 27 participants for the Update Summarization Task. Each participant submitted up to two runs, with run IDs (as shown in Figure 5) from 4 to 55. In addition, three baseline runs were included in the evaluation, which occupy the first three run ID slots (1–3). Baseline 1 returns all the leading sentences in the most recent documents. Baseline 2 returns a copy of one of the model summaries for the document set, but with the sentences randomly ordered. Baseline 3 returns a summary consisting of sentences that have been manually selected from the document set. We submitted two runs: the results in which sentence order is decides utilize chronological information, and the results of our proposed method. The ID for our proposed method is 18; the other one is 44.

Table 1 shows the NIST evaluation results. The evaluation for set B is worse for set A in both runs; to get the update summary, we just avoided including the same information (words) that appears in both sets into the update summary; this method proved very ineffective. With both evaluation method, the results for the two runs are not vary much.

Run 18

ABC News anchor Peter Jennings has been diagnosed with lung cancer, the network announced Tuesday, and will immediately begin a round of outpatient chemotherapy that is expected to keep him from the anchor desk at times in the coming months. ABC News emphasized that Jennings would remain its evening news anchor. While still in his 20s, Jennings anchored ABC's evening news for two years in the 1960s.

Run 44:

E-mail messages released Tuesday by Jennings and ABC News president David Westin stressed that Jennings, who is 66 and who has anchored ABC's "World News Tonight" since 1983, will continue to work during treatment, provided he feels well enough. ABC News emphasized that Jennings would remain its evening news anchor. While still in his 20s, Jennings anchored ABC's evening news for two years in the 1960s.

Figure 4: Summary examples

NIST evaluators also assigned an overall linguistic quality score to each of the automatic and human summaries. The score is guided by consideration of the

Table 1: NIST evaluation results

Document Set	Run 18			Run 44		
	Rouge2	Linguistic quality	Overall responsiveness	Rouge2	Linguistic quality	Overall responsiveness
Set A	0.08009	4.659	4.136	0.08163	5.068	4.114
Set B	0.05252	5.205	3.114	0.04290	4.386	2.795
Average	0.06631	4.932	3.625	0.06227	4.727	3.455

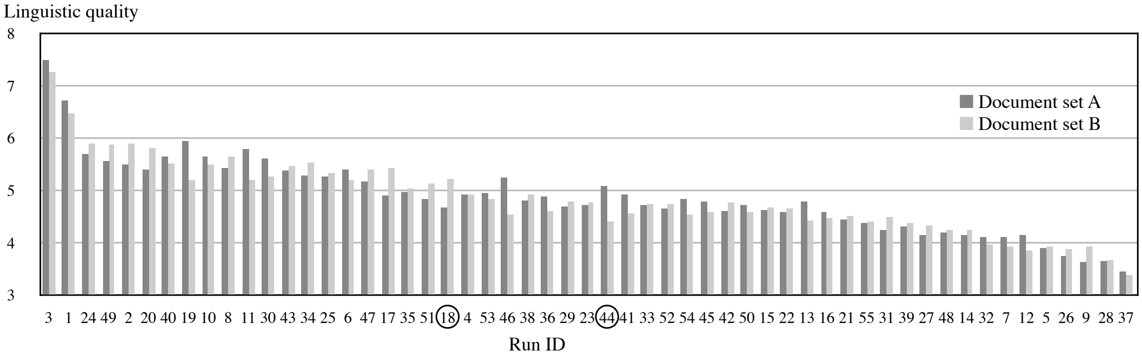

Figure 5: Linguistic Quality Evaluation for set A and set B

Table 2: Number of leading sentences in source as a leading sentence in summary

	Run 18	Run 44	Total
Leading sentence	28	9	40

following factors with integer scores between 1 (very poor) and 10 (very good).

- Grammaticality
- Non-redundancy
- Referential clarity
- Focus
- Structure and Coherence

In the manual evaluation, the linguistic quality and overall responsiveness is different; this to some extent proves the sentence ordering does have an affect on the quality of the summary. As shown in Table 1, the overall linguistic quality for our two runs do not vary substantially. But our proposed method does perform better on document set B.

Figure 5 shows the linguistic quality evaluation results; each group has two scores; the black one is for set A; the grey one is for set B. Our two runs, 18 and 44, ranked 21st and 29th, respectively. Figure 4 shows examples of summaries generated by our two runs; the one from run 18 puts a leading sentence (italic part) in the source document as the leading sentence in the summary which seems more sensible. The leading sentence always has some special "features"; when we order the sentences in the summary, if a sentence contains more features of that kind, it should be used as the leading sentence of the summary. As shown in Table 2, there are 40 draft summaries that contain at least one leading sentence from source documents, and run 44 only use 9 of them as a leading sentence, while run 18 uses 28.

There are some difficulties with our approach. First, during sentence clustering, it is not easy to cluster sentences properly. Second, noise makes it hard to adjust the structure of the binary tree. The evaluation results for the content of set B is not good, while the linguistic evaluation result is acceptable. This may be because after we punish a word's weight for the same features also appearing in set A, it is easier to adjust the order of the sub-tree. This situation is similar with the majority ordering method where each sentence can be seen as a subtopic.

There are some aspects for future improvement. First, noise reduction is a key step in sentence ordering; we may use some clustering method to filter noisy features. Second, the features of the current experiment are single words, we may try some word patterns as features. Third, sentences are ordered based on only the input source documents without any extra sources; we may try to combine some machine learning methods when there is training data available.

4 Conclusion and Future Work

This paper presented the TITech summarization system participating in TAC2009. We discussed our system for the Update task. We proposed an approach to sentence ordering that tries to utilize two heuristics together to create a more readable summary. In our experiments, we investigated the effect of our method for sentence ordering and the influence of sentence ordering on the quality of the resultant summary. In future research we plan to try other binary tree building and adjusting methods. Our final goal is to integrate our summarizer into a natural language processing system capable of searching and presenting web documents in a concise and coherent form.

References

Jin Zhang, Hongbo Xu, Xiaolei Wang, Huawei Shen, Yiling Zeng 2007. *ICT CAS at DUC 2007.* DUC.

Xiaojun Wan, Jianwu Yang 2006. *Improved Affinity Graph Based Multi-Document Summarization.* Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 181–184

Bird, S., and Loper, E. 2004. *NLTK: The natural language toolkit.* In Proc. of 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04).

Kathleen McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. *Towards multidocument summarization by reformulation: Progress and prospects.* AAAI/IAAI, pages 453–460.

C.Y. Lin and E. Hovy. 2001. *Neats:a multidocument summarizer.* Proceedings of the Document Understanding Workshop(DUC).

Regina Barzilay, Noemie Elhadad, and Kathleen McKeown. 2002. *Inferring strategies for sentence orderingin multidocument news summarization.* Journal of Artificial Intelligence Research, 17:35–55.

Naoaki Okazaki, Yutaka Matsuo, and MitsuruIshizuka. 2004. *Improving chronological sentence ordering by precedence relation.* In Proceedings of 20th International Conference on Computational Linguistics (COLING 04), pages 750–756.

McKeown K., Barzilay R. Evans D., Hatzivassiloglou V., Kan M., Schiffman B., &Teufel, S. 2001. *Columbia multi-document summarization: Approach and evaluation.* In Proceedings of DUC.

Barzilay, R N. Elhadad, and K. McKeown. 2002. *Inferring strategies for sentence ordering in multidocument news summarization.* Journal of Artificial Intelligence Research, 17:35–55.

Mirella Lapata. 2003. *Probabilistic text structuring: Experiments with sentence ordering.* Proceedings of the annual meeting of ACL, 2003., pages 545–552, 2003.

Danushka Bollegala, Naoaki Okazaki, Mitsuru Ishizuka. 2006. *A Bottom-up Approach to Sentence Ordering for Multi-document Summarization.* Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 385–392

Nie Yu, Ji Donghong and Yang Lingpeng. 2006. *An adjacency model for sentence ordering in multidocument.* Asian Information Retrieval Symposium(AIRS2006), Singapore., Oct. 2006.

Ji Donghong, Nie Yu 2008. *Sentence Ordering based on Cluster Adjacency in Multi-Document Summarization.*

Jaime Carbonell and Jade Goldstein. 1998. *The use of MMR, diversity-based reranking for reordering documents and producing summaries.* In Proceedings of SIGIR, pages 335–336, Melbourne, Australia.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. *Multidocument summarization by sentence extraction.* In Proceedings of the ANLP/NAACL Workshop on Automatic Summarization, pages 4–48.