Proceedings of 2012 4th International Conference on Machine Learning and Computing
IPCSIT vol. 25 (2012) © (2012) IACSIT Press, Singapore

An Improved Approach to Sentence Ordering For Multi-document Summarization

Tiedan Zhu[1+], Xinxin Zhao [2]

[1] Beijing Institute of Technology

[2] Beijing Institute of Computer Application Technology

Abstract. While sentence ordering for single-document summarization can be determined from the ordering of sentences in the input article, this is not the case for multi-document summarization where summary sentences may be drawn from different input articles. In this paper, we propose the logical-closeness criterion, which can be used to measure the similarity between two sentences. Based on the logical-closeness, we propose an improved agglomerative algorithm to arrange the order of sentences. Evaluation of our augmented algorithm shows an improvement of the ordering over other baseline strategies.

Keywords: Sentence Ordering, Multi-document Summarization, Natural Language Processing

1. Introduction

Automatic Text Summarization (ATS) is an important subfield of the Nature Language Processing (NLP). ATS is defined as “a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that" [1]. The researches on the automatic summarization are mainly concerned on the extracting method of the sentences recently. The order of sentences is seldom considered. In single document summarization, the summary is created from only one document, so that we can arrange the order of the sentences according to the order of sentences in the original document. But in multi-document summarization, sentences are extracted from different documents. So we need to find a strategy to arrange the order of sentences.

Some experiments have shown that the order of the sentences has much influence on the comprehension of a summary. if a document’s sentences were presented in a random order, a great deal of meaning would be lost, or at least convoluted. For example, when Barzilay asked people to order 8 sentences into a summary, 50 people created only 21 unique orderings, with significant overlap. While there are a variety of coherent orderings, this result suggests that the set of coherent orderings is a small subset of all possible orderings [3].

Bollegala presented an approach to arrange the order of sentences based on four criterions [2]. In this paper, we will propose the fifth criterion, which is called logical closeness. The experiments show that our improved approach using logical closeness performs better.

The paper is organized as follows. In section 2 we present related work to this study. In section 3 we describe logical closeness. The approach of ordering sentences using logical closeness is introduced in section 4. An illustrative experiment is provided in section 5. Finally, Section 6 presents our conclusion.

2. Related Works

The traditional method of arranging the order of sentences is the Chronological method. In the research of automatic summarization, most corpus have the time stamp, especially the news corpus. So we can

[+] Corresponding author. Tel.: + 18610217610.
E-mail address: llztd@bit.edu.cn.

arrange the order according to the chronology. The experiments show that the chronological ordering of sentences is an effective heuristic for multi-document summarization [4][5].

Barzilay proposed an improved version of chronological ordering by first grouping sentences into sub-topics discussed in the source documents, then arranging the sentences in each group chronologically [3]. Okazaki proposed an algorithm to improve the chronological ordering by resolving the presuppositional information of extracted sentences [6].

Barzilay and Lee proposed content models to deal with the topic transition in domain specific text. The content models are implemented by Hidden Markov Models (HMMs), in which the hidden states correspond to topics in the domain of interest and state transitions capture possible information-presentation orderings [7]. Ji and Pulman proposed a sentence ordering algorithm using a semi-supervised sentence classification and historical ordering strategy [8].

Bollegala proposed a sentence ordering strategy using four criterions: Chronology criterion, topical-closeness criterion, precedence criterion and succession criterion [2]. The strategy performs better than many baselines. In our research, we found that the topical-closeness criterion was not strong enough to measure the conherece of two sentences. So we proposed the logical-closeness.

3. Logical-Closeness

3.1. Notation Definition

Definition 1: the arrow '→'. For two sentences a and b, we define notation a→b to represent that a and b are adjacent. The 'adjacent' here has two meanings: first, a and b are coherent enough to be connected together in a document (summary); second, a precedes b.

Definition 2: the sentence-chain. A sentence-chain is made up of a chain of adjacent sentences, i.e. A=(a1→a2→⋯→an-1→an) is a sentence-chain with the length of n, where ai is a sentence.

The notation arrow can also be used to sentence-chains, which means the two sentence-chains are adjacent. For two sentence-chains A= (a1→a2→⋯→an-1→an) and B=(b1→b2→⋯→bm-1→bm), A→B= (a1→a2→⋯→an-1→an→b1→b2→⋯→bm-1→bm). The result of two sentence-chains connected with an arrow is still a sentence-chain.

3.2. Definition of Logical-closeness

There are two kinds of closeness among sentences: topical-closeness and logical-closeness. The topical topical-closeness means the closeness in words. Some words in two sentences are completely same or similarity (for example, through the dictionary). The logical-closeness means the closeness in meanings. Perhaps there is even no one same word in two sentences, but they are coherent in logic.

For example, there are two sentences as follows:

(1) It's raining.

(2) The clothes outside should be taken back.

There are no same words in the two sentences above. Through the dictionary, we can't find any relation between 'rain' and 'clothes'. From topical-closeness criterion, the two sentences above are not coherent. But they are coherent in fact, since the logical-closeness between them.

Suppose we have another sentence as follows:

(3) The clothes will get wet in rain.

Sentence (3) is some kind of background knowledge that everybody knows it. If we put sentence (3) between (1) and (2), the topical-closeness will work. The same word 'rain' in (1) and (3), and the same word 'clothes' in (3) and (2). Unforunately, we have no sentence (3) in most cases. We must find a way to measure the logical-closeness.

3.3. Measure of Logical-closeness

In automatic text summarization, each sentence is extraced from the original document. In the original document, there is certain of context around each sentence. So we can learn the logical-closeness from the original document.

Assume there are two documents, Da and Db. We extract sentence a from document Da and sentence b from document Db to constitute the summary. In document Da, sentence a1 is the one just in front of a, and sentence a2 is the one just behind a. In document Db, sentence b1 is the one just in front of b, and sentence b2 is the one just behind b. So in Document Da, a1→a→a2. In document Db, b1→b→b2.

Sentece a1 and a2 are much coherent with sentence a, since they are adjacent in Document Da. If sentece b is similary to a1 or a2, we can consider that sentence a and b are coherent. In the same way, if sentece a is similary to b1 or b2, we can also consider that sentence a and b are coherent. So we can define the logical-closeness measure function as follows:

$$f_{\log ic}(a \to b) = \max(sim(a_1, b), sim(a_2, b), sim(a, b_1), sim(a, b_2)) .$$

Here, sim(a,b) is the similarity of sentence a and b. From the function of f_{logic}, we can see that if sentence a and b1 are similar each other, even completely same, f_{logic} will get a much high value. For example, sentence a is “It’s raining” and sentence b is “The clothes outside should be taken back”. While sentence b1 is “It’s raining”, too. Sim(a,b1) will get a high value so that f_{logic}(a→b) will get a high value. So f_{logic}(a→b) can measure the logical-closeness between sentence a and b.

For sentence-chain A and B, the measure of logical-closeness can be defined as follows:

$$f_{\log ic}(A \to B) \sum_{b \in B} \max_{a \in A} f_{\log ic}(a \to b) .$$

For each sentence $b \in B$, we caculate all the logical-closeness between b and $a \in A$, and choose the maximum as the logical-closeness between A and b. The logical-closeness between A and B can be calculated from averaging all the logical-closeness between A and $b \in B$.

4. Ordering Algorithm

Our goal is to build a function to measure the adjacent relation between two sentence-chains, i.e.

$$f(A \to B) = \begin{cases} p & \text{if A and B are adjacent} \\ 0 & \text{if A and B are not adjacent} \end{cases}$$

Here 0≤p≤1, it represents the probability of A→B.

The definition of f(A→B) can be seen as a classifier of two classes: A and B are adjacent or not. We can use support vector machine (SVM) to accomplish this deal.

Bollegala proposed a sentence ordering strategy using four criterions: Chronology criterion, topical-closeness criterion, precedence criterion and succession criterion. We proposed logical-closeness above. So we can create a five-dimensionality space with the five criterions. For two sentence-chains A and B, we can get a vector in the space, that is (f_{logic}(A→B), f_{topic}(A→B), f_{chro}(A→B), f_{pre}(A→B), f_{succ}(A→B)).

To train a SVM classifier, we need some positive and negative samples, which can be found from summaries created by humans. We select adjacent sentence-chains as positive and non-adjacent sentence-chains as negative samples.

Tab.1: sentences ordering algorithm

sentences ordering algorithm

```
S={A1,A2,⋯,An}
while |S|>1 do
{
  $(A_a, A_b) = \arg\max_{A_i, A_j \in S, A_i \neq A_j} f(A_i \to A_j)$
  $A_a = A_a \to A_b$
```

```
    $S = S - (A_b)$
}
return S
```

We can judge two sentence-chains are adjacent or not by a trained SVM. If A and B are not adjacent, f(A→B)=0; if A and B are adjacent, the distance from vector (f_{logic}(A→B), f_{topic}(A→B), f_{chro}(A→B), f_{pre}(A→B), f_{succ}(A→B)) to the hyperplane can be caculated. We use sigmoid functions to convert the distance into a posterior probability.

Given S={a1,a2,…,an}, firstly we regard S as a set of sentence-chains, in which each sentence-chain contains only one sentence, i.e. S={A1,A2,…,An}={(a1),(a2),…,(an)}. By function (1), we can find the two most adjacent sentence-chains. We connect the them into one sentence-chain. Thus S is a set of n-1 sentence-chains. Repeat this process, until there is only one sentence-chain in S. Then the only sentence-chain is the result of sentence ordering. Tab 1 shows the algorithm.

5. Experiments

5.1. Datasets

The Text Analysis Conference (TAC) and its predecessor the Document Understanding Conference (DUC) hold annual conferences and adjoining competitions to encourage research in automatic multi-document summarization of news articles. Because of the extremely useful aggregation of data and reference summaries that are provided by these conferences, the associated datasets have become the standards in the ATS literature. We present results in our study on the DUC 2006. The DUC 2006 datasets contain 50 document sets of different topic and each topic contains 25 news documents. We choose 30 of them as training datasets and 20 as testing datasets.

5.2. Baselines

We compare the proposed improved agglomerative ordering algorithm (iAGL) with five other sentence ordering algorithms: agglomerative algorithm(AGL), random ordering (RND), human-made ordering (HUM), chronological ordering (CHR), topical-closeness ordering (TOP).

Agglomerative ordering (AGL) is an ordering arranged with four criterions.

Random ordering (RND) is the lowest anchor, in which sentences are arranged randomly.

Human-made ordering (HUM) is the highest anchor, in which sentences are arranged by a human experts.

Chronological ordering (CHR) arranges sentences with the chronology criterion.

Topical-closeness ordering (TOP) arranges sentences with the topical-closeness criterion.

5.3. Experiment Process

It is very difficult to score an order of sentences. There are some automatic methods to do this. But they are not comprehensive or mature. So we ask 5 human experts to score the order of sentences. There are 4 grades: perfect, acceptable, poor, unacceptable. A perfect summary is a text that we cannot improve any further by reordering. An acceptable summary is one that makes sense, and is unnecessary to revise even though there is some room for improvement in terms of its readability. A poor summary is one that loses the thread of the story at some places, and requires minor amendments to bring it up to an acceptable level. An unacceptable summary is one that leaves much to be improved and requires overall restructuring rather than partial revision.

Tab 2, the results of the experiments

	Perfect	Acceptable	Poor	unacceptable
RND	0	5	8	87
TOP	9	23	6	62

CHR	21	25	32	22
AGL	24	39	21	16
iAGL	28	44	20	8
HUM	82	15	2	1

5.4. Results

We have 20 testing datasets and 5 human experts to score the order. So we get 20*5=100 scores for each algorithm. Tab 2 shows the score results.

Fig 2 shows the bar chart of the data in Tab 2. We can see the quality of each algorithm more directly.

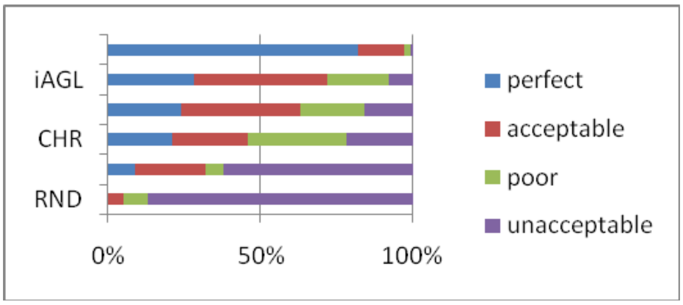

Fig.2 the bar chart of the results

From the results of experiments, we can see that out method proposed here is better than the AGL and much better than other automatic methods. That means the logical-closeness is effective and our method can improve the result of sentence ordering.

6. Conclusion

In multi-document summarization, sentences are extracted from different documents. We cannot arrange the order like single-document summarization. So we need to find a strategy to arrange the order of sentences in multi-document summarization. In this paper, we proposed the logical-closeness of two sentences. We analyse the characteristic of logical-closeness and find the measure of logical-closeness. We propose an improved agglomerative algorithm with logical-closeness. The experiments shows that our method takes effect and is better than other automatic methods.

7. References

[1] Inderjeet Mani. 2001. Automatic Summarization. John Benjamins Publisher.

[2] D. Bollegala, N. Okazaki, and M. Ishizuka. A bottom-up approach to sentence ordering for multi-document summarization. Information Processing & Management, 46(1):89–109, 2010.

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

[4] Lin, C., Hovy, E., 2001. Neats:a multidocument summarizer. Proceedings of the

[5] Document Understanding Workshop(DUC).

[6] Okazaki, N., Matsuo, Y., Ishizuka, M., 2004. Improving chronological sentence ordering by precedence relation. In: Proceedings of 20th International Conference on Computational Linguistics (COLING 04). pp. 750 – 756.

[7] Barzilay, R., Lee, L., 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In: HLT-NAACL 2004: Proceedings of the Main Conference. pp. 113 – 120.

[8] Ji, P. D., Pulman, S., 2006. Sentence ordering with manifold-based classification in multi-document summarization. In: Proceedings of Empherical Methods in Natural Language Processing. pp. 526 – 533.