Holographic Lexical Chain and Its Application in Chinese Text Summarization

Shengluan Hou[1,2], Yu Huang[1,2], Chaoqun Fei[1,2], Shuhan Zhang[1,2], and Ruqian Lu[1,3]

[1]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China

houshengluan1989@163.com
[3] (Key Lab of MADIS, Academy of Mathematics and Systems Science) of Chinese Academy of Sciences, Beijing, China

Abstract. Lexical chain has been widely used in many NLP areas. However, when using it for Web text summarization, especially for domain-specific text summarization, we got low accuracy results. The main reason is that traditional lexical chains only take nouns into consideration while information of other grammatical parts is missing. We introduce lexical chains of predicates and adjectives (adverbs) respectively. These three types of lexical chains together are called holographic lexical chains (HLCs), which capture most of the information included in the text. A specifically designed construction method for HLC is presented. We applied HLC method to Chinese text summarization and used machine learning methods whose features are adapted to the new method. In a comparative study of Chinese foreign trade texts, we got summarization results with accuracy of 86.88%. Our HLC construction method obtained improvements of 7.02% in accuracy than the known best methods in Chinese text summarization.

Keywords: holographic lexical chain, text summarization, machine learning, lexical cohesion.

1 Introduction

Lexical cohesion is a classical tool for analyzing the content of natural language text. Lexical chain is a list of “about the same thing” words, which exploit the lexical cohesion among related words and contributes the continuity of text meaning [13, 19]. Lexical chain has been widely used in many natural language applications, such as text summarization [1-4, 10], machine translation [24], discourse quality measurement [20] and some other areas [5,17, 22].

The influx of large amount of web documents in the web age results in a great deal of attention on automatic text summarization [14, 15]. Lexical chain is a good tool for text summarization because of its easy computation and high efficiency. There are many efforts of lexical chain based text summarization [2, 4, 10, 19]. Generally, these

adfa, p. 1, 2011.

lexical chain based text summarization methods can be decomposed into two procedures: (1) Lexical chains construction; (2) Select sentences relating to lexical chains as summary.

In the first procedure, several lexical chains will be computed from each article. Barzilay and Elhadad described the procedure of lexical chain construction as composed of three steps [2]:

1. Select candidate words, including nouns and named entities;
2. For each candidate word, find an appropriate chain to insert, relying on a relatedness criterion between candidate word and members of the lexical chains;
3. If found, insert the word into the chain and update it accordingly; If not, construct a new chain and insert the word into the new chain.

However, when using it for Web text summarization, especially for domain-specific text summarization, we got low accuracy results. The main reason is that traditional approaches for lexical chain construction only take nouns (and named entities, as shown in [2]) into consideration, while information of other grammatical parts, such as predicates or adjectives, which carry important information for text summarization, is missing. On the other hand, these methods adopt the assumption that “one sense per discourse”, which had been proved not valid by other researchers [9].

The novelty of our approach is to introduce other two kinds of lexical chains, namely the predicates lexical chains and adjectives (adverbs) lexical chains. In this way we have lexical chains for all three major grammatical parts. These three types of lexical chains together are therefore called holographic lexical chains (HLCs for short), which capture most of the information included in the text. A specifically designed construction method for HLCs is presented. Moreover, we present a scoring criterion to identify strong chains and weak chains.

To illustrate this, let us consider the following text with seven sentences:

Rail freight giant Aurizon has cancelled its $91 million effort to standardize legacy systems onto a single SAP platform after an internal review found the project was at risk of delays and overspend.
In early 2014 the company revealed its plan to consolidate 18 separate legacy systems for logistics, planning, scheduling, ordering, and billing onto SAP HANA and SAP’s supply chain execution (SCE) platform 9.1.
The rail operator at the time said the system would improve visibility across the supply chain by "integrating long and short-term planning with resource availability and customer demand".
It had previously implemented SAP for ERP and asset maintenance, and called the HANA and supply chain platform a logical extension.
However, after three years the project will now be cancelled and an impairment charge of $64 million recorded in the company's first quarter FY17 results, Aurizon said today.
Around $27 million of the project's total $91 million cost remains capitalized for software and licenses that are still in use, it said.
Aurizon CEO Andrew Harding said the project was not delivering value for the business.

Fig. 1. An example of Web text

The HLCs can be identified as:

Noun lexical chains:

rail[1], rail[3];

Aurizon[1], Aurizon[5], Aurizon[7];
million[1], million[5], million[6], million[6];
legacy system[1], SAP platform[1], legacy system[2], SAP HANA[2], SAP[2], platform[2], system[3], SAP[4], ERP[4], HANA[4], platform[4], software[6];
project[1], project[5], project[6], project[7].

Predicate lexical chains:

cancel[1], cancel[5];
reveal[2], say[3] say[7].

Adjective(Adverb) chains:

single[1], separate[2];
long[3], short[3].

Fig. 2. An example of HLCs

In Fig. 2, the digits in square brackets are sentence numbers.

We further investigate the connections between text summarization and lexical cohesion. We apply HLC to Chinese text summarization. Our Chinese text summarization method can be viewed as two-step process, too. The first step is to build HLC of a text. The second step is selecting summarizing sentences using machine learning methods whose features are adapted to our new HLC construction method.

Our aim is to:

1. Design an accurate and efficient HLC construction algorithm, which can process documents of any length;
2. Design a robust Chinese text summarizer based on HLC approach together with machine learning techniques.

The rest of this paper is organized as follows. In the next section, we survey the related works about lexical chain construction and text summarization. Section 3 defines and illustrates the concept of HLC. A novel HLC construction algorithm and scoring criterion to identify strong chains and weak chains are also presented in this section. Section 4 illustrates HLC's application in Chinese text summarization. The experiments setup and experimental results analysis are shown. Finally, we conclude the paper with hints on future work in Section 5.

2 Related Works

Morris and Hirst first introduced the concept of lexical chain and showed it can be an indicator of the whole text structure [13]. They used Roget's thesaurus as the major knowledge base for computing lexical chains. Since there was no computer-readable thesaurus at that time, their algorithm cannot be implemented on the computer.

Hirst and St-Onge presented the first computer-implemented lexical chain construction algorithm, which used WordNet [12] as knowledge base [7]. They defined three kinds of word sense relations. When inserting a candidate word into an existing chain, an extra-strong relation is sought throughout all chains. If a chain is found, the word is inserted with the appropriate sense and the senses of other words are updated; If not, strong relation is preferred to medium-relation to sought. Note that there is no limit in distance for extra-strong relations, seven and three sentences are limited for strong and medium-strong relations respectively.

However, Hirst and St-Onge's "greedy disambiguation" resulted in low word sense disambiguation (WSD) accuracy, since their method disambiguates a word at its first encountering. Barzilay and Elhadad proposed a new method according to all possible alternatives of word senses and chose the best one among them [2]. They also used WordNet as knowledge base. This method significantly improves WSD accuracy but with exponential complexity. It constructs all possible interpretations of candidate words and selects that interpretation with the strongest cohesion, which causes the inefficiency.

Based on Barzilay and Elhadad's work, Silber and McCoy defined a linear time algorithm [19]. They rewrote Wordnet noun database and tools to facilitate efficient access, but still with low accuracy in WSD.

Galley and McKeown further investigated automatic lexical chains construction [6]. They separated WSD and lexical chain construction into two different sub-tasks and adopted the assumption of "one sense per discourse". All word semantic links were linked to a disambiguation graph in time $O(n)$ and all wrong links are removed from this disambiguation graph.

Remus and Biemann presented three knowledge-free methods for lexical chain construction [18]. They used the Latent Dirichlet allocation (LDA) topic model for estimating the semantic closeness of candidate terms. These corpus-driven methods adopt the idea of interpreting lexical chains as clusters and a particular set of lexical chains as a clustering. Other lexical chain applications are mostly based on Galley and McKeown's methods.

The above lexical chain methods only take nouns into consideration. Few efforts focused on verb or adjective lexical chains. Novischi and Moldovan used WordNet to create two types of lexical chains for question answering. Verb lexical chains were used for propagating verb arguments, and noun lexical chainss were used to link semantically related arguments [16]. Jarmasz and Szpakowicz constructed lexical chains using Roget's thesaurus, whose hierarchical structure allow them to build lexical chains using nouns, adjectives, verb, adverbs and interjections [8].

Text summarization is a difficult but important research area in the web age of information explosion. Generally, the methods can be categorized into two approaches: abstract-based approaches and extraction-based approaches. Abstract-based approaches can be considered as content synthesis of source text, which needs deep natural language understanding. Extraction-based approaches extract important sentences to generate a concise and coherent version of original text. Extraction-based approaches are main approaches for text summarization.

Lexical chain based text summarization is a method of extraction-based summarization. Many efforts of lexical chain based text summarization have been developed to date. Barzilay and Elhadad presented a method of scoring chains. The sentences containing these strong chains are chosen as summary sentences [2]. Ercan and Cicekli implemented Galley's lexical chain construction method and represented topics by sets of co-located lexical chains to take advantage of more lexical cohesion clues [4].

They segmented text into topic episodes by means of lexical chain clustering. The final summary was sentences selected from each topic segments.

3 Holographic Lexical Chains

Since Morris and Hirst first proposed the concept of lexical chain, many lexical chain construction methods along with lexical chain based applications have been presented to date. However, these lexical chain construction methods only take nouns (and/or named entities) into consideration, while information of other grammatical parts, such as predicates or adjectives, which often carry important information for text summarization, is neglected.

We introduce other two kinds of lexical chains: predicates lexical chains and adjectives (adverbs) lexical chains in this section. Nouns, predicates and adjectives (adverbs) form the major grammatical parts of a sentence. Their corresponding lexical chains capture most of the information included in the text, which we call holographic lexical chains.

The candidate words of holographic lexical chains contain three parts: noun candidate words, predicate candidate words and adjective(adverb) candidate words. All the words were stemmed first. We selected all nouns and named entities as noun candidate words, which describe the topics of a text. For each sentence, there's only one predicate, which describes the property that a subject has and can be recognized as the root of dependency tree. All predicates from each sentence constitute predicate candidate words. All adjectives and adverbs were selected as adjective(adverb) candidate words, which qualify nouns.

Definition 1 Holographic Lexical Chains (HLCs)

Holographic lexical chains of a discourse include three types of lexical chains. They can be summarized with a triple:

$$HLCs = <\mathrm{N}Chains, \mathrm{P}Chains, \mathrm{A}Chains> \tag{1}$$

Where

(1) $\mathrm{N}Chains$ are noun lexical chains, each of which is a list of semantically related nouns, whose elements consist of nouns or named entities;

(2) $\mathrm{P}Chains$ are predicate lexical chains, each of which is a list of semantically related predicates. The elements in each chain are usually verbs;

(3) $\mathrm{A}Chains$ are adjective (adverb) lexical chains, each of which is a list of semantically related adjectives and/or adverbs.

Let $\mathrm{N}Chains \cup \mathrm{P}Chains \cup \mathrm{A}Chains = \{C_1, C_2, \cdots, C_k\}$, where each $C_i (1 \le i \le k)$ is a lexical chain. We have the following properties:

(1) $C_i \cap C_j = \varnothing$, $1 \le i, j \le k$ and $i \ne j$;

(2) $C_1 \cup C_2 \cup \cdots \cup C_k = C$, C is the set of all candidate words.

Property (1) means there's no common words between two different lexical chains. In other words, every word occurrence can only belong to one lexical chain, there is no overlapping word occurrences. Property (2) means for all candidate words, each word should belong to one chain.

For each source text, the process of HLC construction is shown as follows:

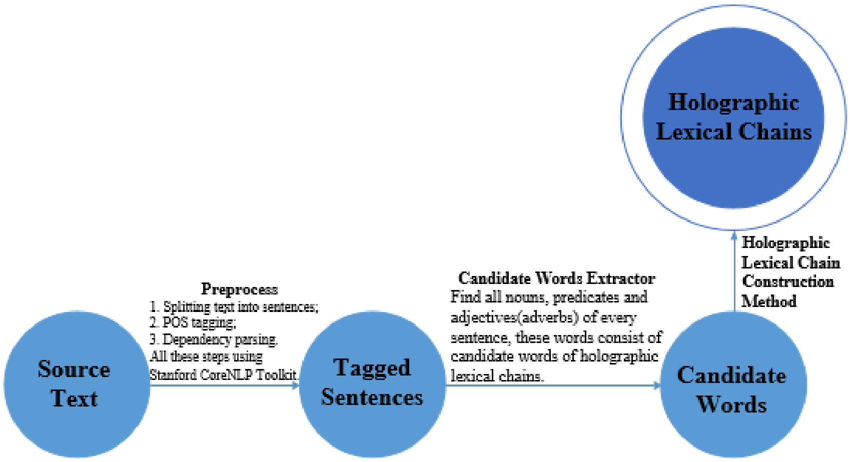

Fig. 3. Process of holographic lexical chains construction

In the preprocessing procedure, we use Stanford CoreNLP Toolkit [11] for sentence splitting, POS tagging and dependency parsing. Then all nouns, predicates and adjectives (adverbs) are extracted, which form candidate words of HLCs. Finally, we present holographic lexical chain construction method to compute HLCs.

Algorithm 1 Holographic lexical chain construction method

Input: Three types of candidate words, denoted by ***cws***.
Output: Holographic lexical chains, denoted by ***hlcs***.

```
nounChains =
    getLexicalChains(cws.nouns);
predicateChains =
    getLexicalChains(cws.predicates);
adjectiveChains =
    getLexicalChains(cws.adjectives);
hlcs.setNounChains(nounChains);
hlcs.setPredicateChains(predicateChains);
hlcs.setAdjectiveChains(adjectiveChains);
```

The function "getLexcialChains" above is a general method for constructing three types of lexical chains. The novelty of our method has two aspects: (1) we adopt the idea of "disambiguation graph" in Galley and McKeown's method but get rid of their assumption of “one sense per word”, different appearances of one word can have different senses; (2) In the word sense disambiguation step, we define three kinds of word sense relations and each with a weight to determine the final word sense.

We choose HIT IR-Lab Tongyici Cilin(Extended version)(means ‘Thesaurus of Synonym Words’, TCE for short) [21] as our knowledge base for automatically qualifying semantic similarity between words. TCE groups words into five levels according to their senses. Each line in the fifth level refers to a common semantic concept, which corresponds to a synset that contains words of similar meaning. Since one word may have more than one sense, a word may occur in more than one synset.

We also define three kinds of word sense relations: extra-strong relation, strong relation and medium-strong relation. They are different from the three kinds of relations in Hirst and St-Onge’s definition. Empirically, we define extra-strong relation as a word and its literal repetition, strong relation as two synonyms words or words with lowest common ancestor level 4 in TCE, and medium-strong as two words belonging to the same word class, where the lowest common ancestor level is 3 in TCE.

The function “getLexcialChains” has the following three steps:

1. Disambiguation graph construction. Disambiguation graph here is an undirected graph, whose vertex denotes word occurrence, and edge denotes the semantic relationship between two word occurrences.
2. Word sense disambiguation. If a word occurrence has more than one senses, find and preserve the most appropriate sense. Remove other senses and their corresponding edges.
3. Final lexical chains building. Traverse disambiguation graph and the words related by edges form one lexical chain. All building lexical chains consist of the final lexical chains.

Algorithm 2 Disambiguation graph construction

Input: Candidate word list, denoted by ***cwl***.
Parameter: The maximum number of sentences between two words who has strong relation, denoted by ***msr***; The maximum number of sentences between two words who has medium-strong relation, denoted by ***mmsr***.
Output: Disambiguation graph.

```
Create an empty disambiguation graph, denoted by dg.
for each word cw_i in cwl
  add cw_i to dg;
  for each word gw_j in dg
    if cw_i and gw_j has extra-strong relation
      create edge between cw_i and gw_j;
    else if cw_i and gw_j has strong relation and
          distance(cw_i, gw_j)<=msr
      create edge between cw_i and gw_j;
```

```
    else if cw_i and gw_j has medium-strong relation
           and distance(cw_i, gw_j)<=mmsr
      create edge between cw_i and gw_j;
    end if
  end for
end for
```

The “distance(cw_i, gw_j)” in Algorithm 2 denotes the number of sentences between the sentence that contains cw_i and sentence contains gw_j. Empirically, we choose the parameters “**msr**” and “**mmsr**” in Algorithm 2 as 7 and 3 respectively.

Algorithm 3 Word sense disambiguation

```
Input: Disambiguation graph, denoted by dg.
Output: Disambiguation graph after word sense
disambiguation.
```

```
for each word cw_i in dg
  if cw_i has more than one sense
    for each sense scw_i^j of cw_i
      compute the weight of scw_i^j;
    end for
    preserve the sense with highest weight, remove
    other senses and their corresponding edges;
  end if
end for
```

In Algorithm 3, the weight of cw_i ’s sense scw_i^j means the sum weight of all edges that scw_i^j related to. If scw_i^j connects to n vertexes $\{v_1, v_2, \cdots v_n\}$ and their corresponding edges are $\{e_1, e_2, \cdots e_n\}$, then the weight of scw_i^j is:

$$W_{scw_i^j} = \sum_{k=1}^{n} W_{relationType} * \frac{1}{distance(cw_i, v_k)} \quad (2)$$

Where

$$W_{relationType} = \begin{cases} 1, \ extra-strong\ relation \\ 0.5, \ strong\ relation \\ 0.3, \ medium-strong\ relation \end{cases} \quad (3)$$

and $distance(cw_i, v_k)$ denotes the number of sentences between the sentence that contains cw_i and sentence contains v_k.

After construction of HLC, there may be some weak chains, such as the chains whose length is 1. One must identify the useful and strong chains. We propose a scoring criterion for chains in HLC. For each chain, the sore can be computed as:

$$Score = ChainStrength * HomogeneityIndex \tag{4}$$

Where

$$ChainStrength = \frac{D_{chainspan}}{N_{all\ sentences}} * L * \sum W_{relationType} \tag{5}$$

and

$$HomogeneityIndex = 1 - \frac{N_{distinctMembers}}{L} \tag{6}$$

The symbol L in formula (5) and formula (6) denotes the number of all words in the chain. In formula (5), $D_{chainspan}$ denotes the number of sentences included in a text block, in which the first and the last sentence contain the first and the last element of the chain respectively; $N_{all\ sentences}$ denotes the number of all sentences of the text; $\sum W_{relationType}$ denotes the sum of all weights of relation between any two words in the chain. $N_{distinctMembers}$ in formula (6) denotes the size of distinct members in the chain. For example, the distinct members in chain "apple, apple, fruit, banana, fruit" are "apple, fruit, banana", whose number is 3.

For each type of lexical chains in HLC, the strong chains satisfy the following formula:

$$Score > AverageScore + \alpha * StandardDeviation \tag{7}$$

The weak chains satisfy the following formula:

$$Weak\ chain = \begin{cases} length < 2, \\ Score < AverageScore - \beta * StandardDeviation. \end{cases} \tag{8}$$

Where $AverageScore$ and $StandardDeviation$ mean the average score and standard deviation of this type of chains belonging to a text, respectively. Here α and β are parameters which can be determined by specific requirements. Empirically, α and β can be determined as 2 by default.

When using HLC to specify applications, weak chains can be removed or only strong chains be selected for better performance and high efficiency.

4 Application in Chinese Text Summarization

In order to verify the validity of our HLC method, we applied it for Chinese text summarization. Since there is no Chinese text summarization benchmark dataset, we chose Chinese foreign trade texts as the domain of our experimental corpus and extracted articles from the Internet. We randomly selected 159 texts as experiment dataset, from which the summary sentences are tagged manually. The average sentence length of these texts is 53.59 and the average number of sentences in each text is 13.53. Excerpts of one of this text is shown in Fig. 4.

> 对于一季度的外贸形势,商务部长高虎城近日表示,对2月份可能出现的外贸数据波动,各界要有充分的思想准备。
> For the foreign trade situation in the first quarter, Minister of Commerce Gao Hucheng said recently that various circles of society should be fully prepared to undertake the probable data fluctuation of foreign trade in February.
> 高虎城指出,春节因素并不是造成中国开年进出口“双降”的唯一原因。
> Gao Hucheng pointed out that the Spring Festival is not the only reason for the "double down" of China's import and export.
> 由于全球市场需求的疲弱及大宗商品价格的低迷,世界主要经济体的进出口均表现不佳,造成了全球贸易开局不佳的态势。
> Due to the weak global market demand and the sluggish bulk commodity prices, the world's major economic entities were the poor performance of imports and exports, resulting in a poor start of global trade trend.

Fig. 4. An example of Chinese foreign trade text(excerpt)

In the summary generation step, we used machine learning methods rather than heuristic rules. We treated summary sentence selection as a binary classification problem. Features used in this step are the following:

a. Co-appearance of words from different types of lexical chains. Since nouns, predicates and adjective (adverb) constitute all major grammatical parts, we took those sentences as summary sentences, which contain words from two or more types of lexical chains. For example, if a sentence contains three types of lexical chain words “进出口(Import-Export), 增长(Rise), 明显(Obviously)”, the sentence may be a summary sentence. There are four vectors of this feature:

Table 1. Four vectors of word co-appearence feature

Noun+ Predicate+Adjective(Adverb)
Noun+ Predicate
Noun+ Adjective(Adverb)
Predicate+Adjective(Adverb)

b. Appearance of the first word of any lexical chain. It means that this sentence is the begin of a new topic, which should be considered as a summary sentence.
c. Appearance of the representative word of any lexical chain. A representative word is the highest frequency word of a lexical chain. It means that the content of this sentence is closely related to the topic of this lexical chain.
d. Appearance of two or more elements of the same lexical chain. It means that this sentence is more closely related to the lexical chain than the others.
e. Appearance of words from more lexical chains than the others.
f. The first sentence of a text.
g. The long sentence. We define long sentence here as:

$$Length > AverageLength + \lambda * StandardDeviation \tag{9}$$

Where λ is a parameter which was taken as 2 in this experiment.

According to these features the summary sentence of example text in Fig.1 is its first sentence.

In our experiment, after HLC construction, the algorithm filters out weaker lexical chains for better performance and high efficiency.

Table 2. HLC filter operation

Chain type	**Filter operation**
Noun chains	Select strong chains
Predicate chains	Remove weak chains
Adjective(Adverb) chains	Remove weak chains

We used both linear regression and support vector machine methods with different feature combinations for comparative study.

To evaluate our method comprehensively, we used four evaluation criteria: accuracy, precision, recall and F1 value, which measure the performance of our method on different aspects.

Table 3. The confusion matrix

All Sentences	Predicted	
	Predicted as summary sentences	Predicted as non-summary sentences
Tagged sentences	TP	FN
Non-tagged sentences	FP	TN

Where TP represents the number of tagged sentences that are correctly predicted as summary sentences; FN represents the number of tagged sentences that are falsely predicted as non-summary sentences; FP represents the number of non-tagged sentences that are falsely predicted as summary sentences; TN represents the number of non-tagged sentences that are correctly predicted as non-summary sentences.

The evaluation criteria are defined as follows.

Accuracy: the percentage of correctly predicted sentences out of all sentences:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

Precision: the percentage of correctly predicted as summary sentences out of all tagged sentences:

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

Recall: the percentage of correctly predicted as summary sentences out of all predicted sentences:

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

F1: the harmonic mean of Precision and Recall, which is calculated as:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{13}$$

We used ten-fold cross validation and computed the average value of those above four criteria.

Table 4. Evaluation results (%) of text summarization

Methods	Features	Average Accuray	Average Precision	Average Recall	Average F1
Support Vector Machine	All	86.88	84.40	73.89	76.47
	All-a	83.67	85.61	60.32	67.35
	a+b+c	80.23	73.21	71.38	69.54
Linear Regression	All	83.31	78.34	69.78	71.51
	All-a	82.19	79.51	58.11	66.93
	a+b+c	79.05	81.23	63.65	69.62

The evaluation results using ten-fold cross validation are shown in Table 4, where “a”, “b” and “c” denote the first three features respectively. “All” denotes all above features from “a” to “g”. “All-a” denotes all above features except “a”. “a+b+c” denotes combination of features “a”, “b” and “c”. The accuracy of 86.88% showed that HLC is a good tool for Chinese text summarization. Furthermore, experimental results confirmed that the feature of word co-appearance plays an important role in Chinese text summarization, which captures most information of a sentence. Support vector machine method outperforms linear regression method.

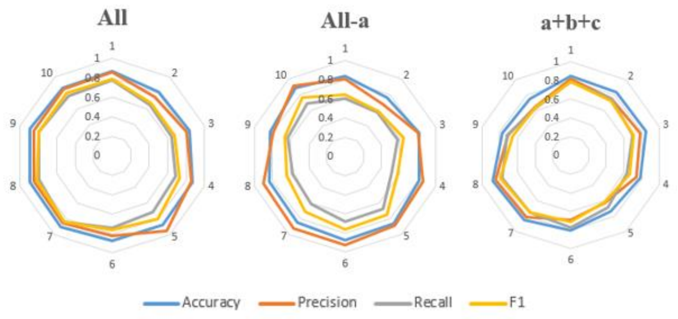

Fig. 5. Data fluctuation of four criteria (support vector machine method)

Table 4 only shows the average values of four criteria, from which we cannot get the data fluctuation information of ten-fold experimental results. We take the results of support vector machine method as an example. Fig.5 shows the data fluctuation of four criteria from ten-fold experiments, from which we can see that the fluctuating range of experimental results is small. Our experimental results of ten-fold cross validation remain stable.

The reasons of failure of identifying the correct summary sentences including two aspects: (1) the tagged sentences were falsely classified as non-summary sentences, because existing features cannot cover all their characteristics; (2) the non-tagged sentences were falsely classified as summary sentences, because these sentences capture common features with summary sentences. According to statistics, the percentage of reason (1) is almost the same as that of reason (2).

For comparison, we re-implemented three main traditional lexical chain construction methods, and used them to construct traditional noun lexical chains on the same Chinese foreign trade corpus. These three methods are listed as follows:

H&S: Algorithm by Hirst and St-Onge [7], with TCE as its knowledge base.

B&E: Algorithm by Barzilay and Elhadad [2], with TCE as its knowledge base.

G&M: Algorithm by Galley and McKeown [6], with TCE as its knowledge base.

Previous lexical chain based text summarization methods used heuristic rules to select summary sentences, such as appearance of the head word of a lexical chain, appearance of representative word of a lexical chain, co-appearance of two or more lexical chains. We used support vector machine method whose features are adapted to these above rules.

Table 5. Comparison (%) of different lexical chain construction methods for text summarition

Algorithm	Accuracy	Precision	Recall	F1
H&S method	76.39	67.37	60.34	60.82
B&E method	79.25	69.85	65.90	63.75
G&M method	79.86	70.21	64.28	64.03
Our method	**86.88**	**84.40**	**73.89**	**76.47**

The accuracy results in Table 5 show the advantage of our innovation in HLC construction method and feature selection outperforms other methods with improvement of 7.02% than the best of them(G&M method).

Meanwhile, we tested the running efficiency of our HLC method and other three traditional lexical chain methods on the same platform. We used a PC with an I5 CPU(2.50 GHz*4 Cores) and 16G memory.

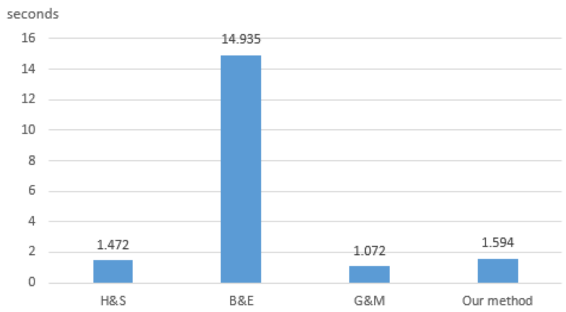

Fig. 6. Time cost of different methods

Our method can construct three types of lexical chains at the same time while the other three methods only construct noun lexical chains. Nevertheless**,** as Fig.6 shows, the time cost of our method is nearly the same as H&S method and G&M method. These three methods are considerably faster than B&E method.

5 Conclusion

In this paper, we presented our approach of holographic lexical chains (HLCs), which contain three kinds of lexical chains: noun lexical chains, predicate lexical chains and adjective (adverb) lexical chains. HLCs capture most of the information included in the text. A specifically designed HLC construction method and scoring criterion are also presented in this paper. We applied HLC technique to Chinese text summarization. In the summary sentence selection step, we used two machine learning methods: linear regression and support vector machine whose features are adapted to HLC and other text features. Comparative experiments on Chinese foreign trade text summarization demonstrate that our holographic lexical chain construction method outperforms other methods in Chinese text summarization.

One of the future works will be further improving efficiency and accuracy of the HLC technique. A possible way is to integrate HLC with some other features, such as those of discourse structure. Another important task is further applying HLC to other NLP applications.

Acknowledgement

This work was supported by National Key Research and Development Program of China under grant 2016YFB1000902, National Natural Science Foundation of China (No. 61232015, 61472412, 61621003), Beijing Science and Technology Project: Machine Learning based Stomatology and Tsinghua-Tencent-AMSS Joint Project: WWW Knowledge Structure and its Application.

References

1. Alam H, Kumar A, Nakamura M, et al. Structured and Unstructured Document Summariza-tion: Design of a Commercial Summarizer using Lexical Chains[C]//ICDAR. 2003, 3: 1147.
2. Barzilay R, Elhadad M. Using lexical chains for text summarization[J]. Advances in automatic text summarization, 1999: 111-121.
3. Brügmann S, Bouayad-Agha N, Burga A, et al. Towards content-oriented patent document processing: Intelligent patent analysis and summarization[J]. World Patent Information, 2015, 40: 30-42.
4. Ercan G, Cicekli I. Lexical cohesion based topic modeling for summarization[C]//International Conference on Intelligent Text Processing and Computational Linguis-tics. Springer Berlin Heidelberg, 2008: 582-592.

5. Feng W L. Research of Theme Statement Extraction for Chinese Literature Based on Lexical Chain[J]. International Journal of Multimedia and Ubiquitous Engineering, 2016, 11(6): 379-388.
6. Galley M, McKeown K. Improving word sense disambiguation in lexical chaining[C]//IJCAI. 2003, 3: 1486-1488.
7. Hirst G, St-Onge D. Lexical chains as representations of context for the detection and correction of malapropisms[J]. WordNet: An electronic lexical database, 1998, 305: 305-332.
8. Jarmasz M, Szpakowicz S. Not as easy as it seems: Automating the construction of lexical chains using roget's thesaurus[C]//Conference of the Canadian Society for Computational Studies of Intelligence. Springer Berlin Heidelberg, 2003: 544-549.
9. Krovetz R. More than one sense per discourse[J]. NEC Princeton NJ Labs., Research Mem-orandum, 1998.
10. Li J, Sun L, Kit C, et al. A query-focused multi-document summarizer based on lexical chains[C]//Proc. of Document Understanding Conference. 2007.
11. Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60.
12. Miller G A. WordNet: a lexical database for English[J]. Communications of the ACM, 1995, 38(11): 39-41.
13. Morris J, Hirst G. Lexical cohesion computed by thesaural relations as an indicator of the structure of text[J]. Computational linguistics, 1991, 17(1): 21-48.
14. Munot N, Govilkar S S. Comparative study of text summarization methods[J]. International Journal of Computer Applications, 2014, 102(12).
15. Nenkova A, McKeown K. A survey of text summarization techniques[M]//Mining text data. Springer US, 2012: 43-76.
16. Novischi A, Moldovan D. Question answering with lexical chains propagating verb arguments[C]//Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2006: 897-904.
17. Qian T, Ji D, Zhang M, et al. Word Sense Induction Using Lexical Chain based Hypergraph Model[C]//COLING. 2014: 1601-1611.
18. Remus S, Biemann C. Three Knowledge-Free Methods for Automatic Lexical Chain Extrac-tion[C]//HLT-NAACL. 2013: 989-999.
19. Silber H G, McCoy K F. Efficiently computed lexical chains as an intermediate representation for automatic text summarization[J]. Computational Linguistics, 2002, 28(4): 487-496.
20. Somasundaran S, Burstein J, Chodorow M. Lexical Chaining for Measuring Discourse Coherence Quality in Test-taker Essays[C]//COLING. 2014: 950-961.
21. Wanxiang Che, Zhenghua Li, Ting Liu. LTP: A Chinese Language Technology Platform. In Proceedings of the Coling 2010: Demonstrations. 2010.08, pp13-16, Beijing, China.
22. Wei T, Lu Y, Chang H, et al. A semantic approach for text clustering using WordNet and lexical chains[J]. Expert Systems with Applications, 2015, 42(4): 2264-2275.
23. Wu X, Xie F, Wu G, et al. PNFS: personalized web news filtering and summarization[J]. International Journal on Artificial Intelligence Tools, 2013, 22(05): 1360007.
24. Xiong D, Ding Y, Zhang M, et al. Lexical Chain Based Cohesion Models for Document-Level Statistical Machine Translation[C]//EMNLP. 2013: 1563-1573.