Chinese Analogy Search Considering Multi Relations

Chao Liang
Department of Computer Science and Technology
East China Normal University
Shanghai, China
chliang@ica.stc.sh.cn

Zhao Lu
Department of Computer Science and Technology
East China Normal University
Shanghai, China
zhu@cs.ecnu.edu.cn

Abstract—There are some specific relations between the entities of the nature. The same relationship exists between two pairs of entities, simply to say, what A is to B as C is to D. For two entities (A,B), we cannot say precisely the relations between them, and given an instance C similar to A, we cannot figure out the entity D which corresponding to B. Analogical search provides a new way of searching D after given the query (A,B) and (C,?). This paper proposed a Latent relation search method based on words frequency and weight of words. We first explore all the relations between two entities A and B using K-means clustering method, then we determine D corresponding to C in various relations. The proposed method achieves an MRR of 0.773.

Index Terms—Analogical search, multiple relationships, K-means, Information retrieval

I. INTRODUCTION

Latent Relational Search is a recently proposed query-by-example technique that aims at solving queries in which the user specifies a triplet of terms (A,B,C) and seeks from a search engine a fourth term D whose relationship with C is analogous to that of A and B. To this reason, we also view Latent Relational Search as Analogical Search.

Kato et al. [1] explained concepts and applications of latent relational search for Japanese in detail. Duc et al. [2] proposed the Latent Relation Search, and a ranking method by employing symmetry property in relational similarity is proposed in [3]. The research in [4] suggested the Cross-Language Latent Relational Search (CLRS). Despite the wide studies of latent relation search, accurately finding the relation between word pair remains a challenging task because of the several reasons. First, relation is a dynamic and varies with time. For example, two companies are competitors original and subsequently one company purchases the other. Second, there is more than one relation in a word-pair. For example, between two words ostrich and bird, besides the relation is a large, there is also the relation is a flightless. A Latent Relation Search measure must extract all relations between them before it can return the target word. Third, there can be more than one way a particular semantic relation can be expressed.

Until now, the three questions mentioned above are not discussed well. Especially the multi relations between a word-pair is not considered in the previous papers. In this paper, three kinds of relation mapping between two word-pairs are first classified, they are one-to-one, one-to-many and multi-relations respectively. We can get different result sets according to three kinds of relation mapping.

The main contributions of this paper include:

(1) We use both bag of words and lexical patterns to represent relations between word-pairs. Frequency and weight of a word are used to rank the candidate relation-words and the target words.

(2) K-means clustering method is used to extract all the relation-words which representing various relations in the word-pair (A,B).

This paper organized as follows. Related works are discussed in Section II. The detailed procedures of extracting relation-words and the target words are shown in Section III. Section IV presents the experimental results and Section VI concludes this paper.

II. RELATED WORKS

Kato et al. [1] introduced latent relation search in detail. Their methods are based on Term Co-occurrence or Lexico-Syntactic Patterns. Relations between two words are represented by bag-of-word in the Term Co-occurrence based method. Another method in their paper treats n-grams as the relation representation between two words. They query a Web search engine for terms or Lexico-Syntactic patterns which in same sentence containing both A and B. The term or pattern set T is supposed to contain terms or patterns which express the relations between A and B. Then, the candidates list of D using C and terms or patterns are extracted. Finally, they rank the candidate list and get the result D. The combination of two approaches gets a highest precision. However, they only extract the words between A and B within a sentence which is not enough for Chinese.

Duc et al. [2] propose the Japanese latent relation search. The proposed approach extracts word pairs from a text corpus (the Web), represents relations between words using lexical patterns which achieving both high precision and recall. However, they need to extract and train a large number of corpus to build a super large index library. In addition, their test sets are defined in limited domains, and they didn’t show precisions for other domains. Duc et al. [4] propose a Cross-Language Latent Relational Search (CLRS). A hybrid (soft/hard) lexical pattern clustering algorithm to recognize paraphrased lexical patterns across languages is put forward. Using the proposed clustering algorithm, they can measure the relational similarity between two entity pairs in different languages and rank the result list of a CLRS query.
The approach put forward in [1] imposes a bias towards the frequency of a word. A high frequency word D has a higher probability of being assigned a top rank, irrespective of the semantic relation shared between (A, B) and (C, D). Goto et al. [3] propose a ranking method which uses symmetry feature in relational similarity to alleviate this phenomenon. They also propose “complementary rank” to improve precision in ranking result set of a relational query.

III. OUR APPROACH

The high-level flow of the Chinese latent relational search is illustrated in Fig.1.

A. Entity and Mention

In this paper, we refer an entity as a set of objects in the world. A mention is a reference to an entity. Entities may be referenced in a text by their name, indicated by a common noun or noun phrase, or represented by a pronoun. For example, the following are several mentions of a single entity:
- Name Mention: Joe Smith
- Nominal Mention: the guy wearing a blue shirt
- Pronoun Mentions: he, him

In traditional, the Chinese noun can be classified into three types which are proper noun, abstract noun and direction noun. The entity that we study is mainly represented by proper nouns. Proper nouns can be classified into four subtypes which representing person, object, place and time. All these nouns can be called mentions in this paper.

B. Three Kind of Relation Mappings

Given a Chinese query, {(A,B), (C,?)}, our aim is to get all candidates of D which are the answer of “?”. We call (A,B) as the example word-pair, A and B are two example words, (C,D) is viewed as the target word-pair, where C is the third word, and D is the target word. In this paper, relation-words are used to describe the relations of a word-pair.

The variety of relations of a word-pair and the diversity of target words is considered in the paper. For example, if A is Earth and B is Moon, the Moon is one of satellites of Earth. If we set Mars as the third word C, then all the satellites of Mars should be returned. That is to say, the relation of the example word-pair is single, while C can be mapped to many D according to this relation. We view this kind of relation as one-to-many mapping. Considering another example, the relations between two persons are changing with time. When they were young, they were classmates. They may become couples or competitors when they grew up. In this example, the relations between the two persons are multiple, and each relation refers to one or many target words.

![Fig. 1. The High-level Flow of the Chinese analogy Search](image-url)

Given a query, {(A,B), (C,?)}, if the example word-pair has only one relation, and the number of target words for the relation is one, we call this kind of mapping as One-to-One mapping for Single Relation (OTOSR), shown in Fig.2(a). For
OTOSR, a certain target word will be returned similar to existing methods. If the example word-pair has one relation and the number of target words corresponding to the relation is many, we called this kind of mapping as One-to-Many mapping for Single Relation (OTMSR), shown in Fig.2(b). If there are many relations between two words, and the number of target word is one or many for each relation, we call this kind of mapping as Multi-relations mapping (MR), shown in Fig.2(c).

C. Extract relational words

Our approach starts extracting sentences which contains A and B from a web page. Then obtain list of result that contains A, B through a search engine. We also extract the links in the result list and download the content of each link to obtain more corpus.

Chinese words can be broadly divided into different types according to the Part of Speech (POS), which are nouns, pronouns, verbs, adjectives, adverbs, conjunctions, prepositions, numerals, quantifiers, and modal particles. Nouns include names of people, place names, proper nouns and common numerals, quantifiers, and modal particles. Nouns include names of people, place names, proper nouns and common numerals. It is obvious that all the part of speech of word in our studied word pairs are nouns and the words representing the relation between word pair are nouns and verbs.

1) Extract relation- words

The content of each crawled page is extracted. And then we extract the complete sentences containing A and B. By manually analyzing a large number of corpora, we summarize the lexical-patterns for extracting relation word R and target word D. For example: 红楼梦(红楼梦)(A Dream of Red Mansions) 作者(作者)曹雪芹(Cao Xueqin), 红楼梦 红楼梦 here are proper nouns, 曹雪芹 is name of people, 作者 is the noun that representing the relationship between 红楼梦 and 曹雪芹. According to this example, we can extract the lexical pattern XRnY, where X and Y for the input word, Rn represents the relation between word pair are nouns and verbs.

Some examples of lexical patterns are listed in the table I.

<table>
<thead>
<tr>
<th>XnY</th>
<th>XRnY</th>
<th>X<em>R</em>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X*RY</td>
<td>X<em>R</em>Y</td>
</tr>
<tr>
<td></td>
<td>X*RY</td>
<td>X<em>R</em>Y</td>
</tr>
<tr>
<td></td>
<td>X*RY</td>
<td>X<em>R</em>Y</td>
</tr>
<tr>
<td></td>
<td>X*RY</td>
<td>X<em>R</em>Y</td>
</tr>
<tr>
<td></td>
<td>X*RY</td>
<td>X<em>R</em>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XY*R</th>
<th>XY*Y</th>
<th>XY*Y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>XY*Y</td>
<td>XY*Y</td>
</tr>
<tr>
<td></td>
<td>XY*Y</td>
<td>XY*Y</td>
</tr>
<tr>
<td></td>
<td>XY*Y</td>
<td>XY*Y</td>
</tr>
<tr>
<td></td>
<td>XY*Y</td>
<td>XY*Y</td>
</tr>
<tr>
<td></td>
<td>XY*Y</td>
<td>XY*Y</td>
</tr>
</tbody>
</table>

In this paper, the relation extraction is based on word frequency and lexical patterns. The frequency of potential relation word is calculated by accumulating the occurrence times of word and the weight of each potential relationship word which is calculated at the same time.

The first step is word segmentation and POS tagging [5]. Check the segmented sentence is whether comply with a lexical pattern. In this process, we record all the lexical patterns where the word appears. These lexical patterns can be used in the process of searching D.

Frequency and weight of each word can be calculated by Equation (1) and Equation (2). The definition of weight is the times of the word which occurs in a sentence that match a lexical pattern.

We denote s, w and p as a sentence, a word and a lexical pattern respectively. And we define P and S as follows,

\[ P = \{ p_1, p_2, \ldots, p_j \} \]
\[ S = \{ s_1, s_2, \ldots, s_i \} \]

\[ f(i) = \begin{cases} 1, & w \in s_i \\ 0, & \end{cases} \]

\[ \text{frequency}(w) = \sum_{i=0}^{n} f(i) \] (1)

\[ \text{we}(i) = \begin{cases} 1, & w \in s_i \text{ and } s \text{ match } p_j \\ 0, & \end{cases} \]

\[ \text{weight}(w) = \sum_{i=0}^{n} \text{we}(i) \] (2)

2) K-means Clustering method

As mentioned above, due to the existence of different relationships between two entities, the words representing relationships between entities are more than one. In order to distinguish the different words on behalf of different relations, we use the K-means clustering to clarify the words into different clusters. The noise words with zero weight are ignored. We treat the word with the highest frequency and the highest weight as a central word. Then we calculate the similarity between other candidate words and the central word respectively by using the Chinese word similarity measure based on How-Net [6] and proposed in Liu and Li’s paper [7].

Frequency and similarity we got above of each word are mapped to a point in the two-dimensional plane. K value is the number of cluster, the initial value of k can be estimated by the number of relation-representing words. We optimize the value of k by experiment. First of all randomly generate k-center point, and then calculate the distances to the center of the k clusters for the remaining points. Add each point to the cluster whose center point has the closest distance with this point. Then calculate the error sum of squares of all points with their error sum of squares equal to the error sum of squares of last iteration, clustering is completed.

The clustering algorithm can be described as below.

\[ X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d \]

the set of point, x is two dimension vector,

\[ C = \{c_1, c_2, \ldots, c_k\} \]
k cluster centers. \( c_i \in R^5, 2 \leq k \leq n \)
\[
u = \{u_{ij}\}_{k \times n} \in M_{cen}
\]
a membership matrix.

K-means algorithm clusters \( n \) vector \( x_i (i = 1, 2, \ldots, n) \) into \( k \) clusters \( G_i (i = 1, 2, \ldots, k) \) and calculate the center of each cluster. Make sure that 
\[
J (C) = \sum_{i=1}^{n} \sum_{k=1}^{K} \|x_i - c_{k(i)}\|^2
\]
get the minimization.

Main steps of the algorithm are follows:

Step 1: Initialization: Generate the initial center.

\[
C^{(0)} = \{c_1^{(0)}, c_2^{(0)}, \ldots c_k^{(0)}\}, l=0, 1 \text{ is the iteration time}, T \text{ is the maximum of iteration time. The threshold is } \varepsilon.
\]

Step 2: \( u_{ij}^{(l+1)} \) is updated by Equation (3),
\[
u_{ij}^{(l+1)} = \begin{cases}1, & \text{if } i = \text{arg} \min \{\|x_j - c_l^{(l)}\|\} \\ 0, & \text{else} \end{cases}
\]

Step 3: \( c_l^{(l+1)} \) is updated by Equation (4),
\[
c_l^{(l+1)} = \frac{\sum_{j=1}^{n} u_{ij}^{(l+1)} x_j}{\sum_{j=1}^{n} u_{ij}^{(l+1)}}
\]

If \( \|c_l^{(l+1)} - c_l^{(l)}\| < \varepsilon \) or \( l < T \), then stop; or \( l = l + 1 \), goto Step 2.

After clustering, we select the word with the highest frequency and the highest weight as the relation-word.

D. Extracting Target Words

Because of the relational similarity between A, B and C, D. We can clearly see that the part of speech of A and C is the same as well as B and D. This is important clue to find D.

First identify all the sentences which include the relation representative word R. We also take (红楼梦,曹雪芹) as an example. For lexical pattern XRNy, C corresponds to X which part of speech is a noun and is also a proper noun, R corresponds to N which part of speech is a noun, D corresponds to Y which part of speech must be a noun and also must be a name of people.

From this example we can see that we only need to extract the word D which part of speech is the same with word B. We also calculate the word frequency and weight according to Equation (1) and Equation (2).

Ultimately, the target word is selected based on the frequency and the weight. For OTMSR case, the target words corresponding to each relation are multiple.

IV. EXPERIMENT EVALUATIONS

According to the part of speech of A, B, entity pairs are divided into the following categories: name of person, name of person; place name, place name; name of person, place name; proper noun, name of person; proper noun, proper noun; proper noun, common noun; common noun, common noun; place name, common noun; name of person, common noun. In accordance with the type of entity pairs, we manually create a set of 25 test groups to evaluate the performance of our approach. Detailed examples of test cases are shown in Table II.

<table>
<thead>
<tr>
<th>ID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Li Yaping</td>
<td>Li Yan</td>
<td>Zhao Benhuan</td>
<td>Zhao Yufang</td>
</tr>
<tr>
<td>7</td>
<td>Lin Xiang</td>
<td>Sun Haining</td>
<td>Lin Dan</td>
<td>Yang Xianhua</td>
</tr>
<tr>
<td>9</td>
<td>Huang Jianping</td>
<td>Huang Jiaoe</td>
<td>Yang Wu</td>
<td>Jiang Wen</td>
</tr>
<tr>
<td>17</td>
<td>Lin Daqiu</td>
<td>Chen Xiaoxing</td>
<td>Shanqg</td>
<td>Yan Huaili</td>
</tr>
<tr>
<td>29</td>
<td>Zhao Benhuan</td>
<td>Xue Shengshang</td>
<td>Hou Yaowen</td>
<td>Gao Deng</td>
</tr>
<tr>
<td>31</td>
<td>Cai Zhiqian</td>
<td>Xu Hao</td>
<td>Cai Shuofen</td>
<td>Xu Yi</td>
</tr>
<tr>
<td>47</td>
<td>Dan Lin</td>
<td>Xu Xianfeng</td>
<td>Yao Ming</td>
<td>Ye Li</td>
</tr>
</tbody>
</table>

TABLE II. EXAMPLES OF TEST CASES CATEGORIZED BY WORD PAIR TYPE

A. Performance of k-means clustering method

There are not many instances of words-pair which type is MR. We take two examples to show the effect of K-means clustering method. By using -means clustering method, different relations are distinguished.

The first example is (France, Paris) and (England,?), there are nine candidate relation-words which are listed in Table III.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Relation word</th>
<th>Frequency</th>
<th>Similarity</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Capital</td>
<td>130</td>
<td>1.00</td>
<td>130</td>
</tr>
<tr>
<td>2</td>
<td>Political Center</td>
<td>30</td>
<td>0.10</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>Tour guide</td>
<td>17</td>
<td>0.05</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Classic</td>
<td>15</td>
<td>0.12</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Culture center</td>
<td>32</td>
<td>0.19</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>Train</td>
<td>20</td>
<td>0.17</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>Traffic</td>
<td>24</td>
<td>0.17</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>Story</td>
<td>22</td>
<td>0.17</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>Snow</td>
<td>20</td>
<td>0.26</td>
<td>20</td>
</tr>
</tbody>
</table>

The second example is (Russian, Putin) and (American,?), there are 24 candidate words. The frequency and weight of the top two relation-words President and Premier is much higher.
than the left 22 candidate words. In our experiment, these two words are clustered into one group which is show in table IV.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Relation word</th>
<th>Frequency</th>
<th>Similarity</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>President</td>
<td>89</td>
<td>1.00</td>
<td>64</td>
</tr>
<tr>
<td>1</td>
<td>Premier</td>
<td>31</td>
<td>1.00</td>
<td>26</td>
</tr>
<tr>
<td>1</td>
<td>Candidate</td>
<td>2</td>
<td>0.90</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>Chairman</td>
<td>1</td>
<td>0.90</td>
<td>1</td>
</tr>
</tbody>
</table>

**TABLE IV. CLUSTERING RESULT OF (FRANCE, PARIS);(ENGLAND,?)**

B. Experiment Results

The accuracy of analogy search is determined by the accuracy of the extracted relation-words. In our experiment, 68 percentages relation-words of questions are extracted correctly. By using the right relation-words, we can almost find out the target words correctly. From Fig.3 we can see that the recall rate of our method is 0.94 and the precision rate is 0.68.

![Fig. 3. Percentage of questions which target words at different rank](image)

The relation-word rank of each test case is showed in Fig.4. In the 19th group, our intention is to find out the famous product of different provinces in China, but unfortunately there are a lot of noises such as the product advertisements. The reason why rank of the 11th and 12th group is low is obvious: when the money is mentioned, a large group of irrelevant information such as exchange rate, policy and bond are involved; it is hard for the engine to find the real relation that the dollar is the currency of US. For the 24th and 44th group, the corpus extracted from the web is limited which lower the accuracy. Based on the above analysis, we conclude that if the corpus has a large amount of noise or the relative phrases are little, their ranks are low.

![Fig. 4. The relation word rank of each test case](image)

**TABLE V. MRR AND PERCENTAGE OF TARGET WORDS AT DIFFERENT RANK**

<table>
<thead>
<tr>
<th></th>
<th>@1</th>
<th>@5</th>
<th>@10</th>
<th>@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS</td>
<td>0.773</td>
<td>68.0</td>
<td>94.0</td>
<td>94.0</td>
</tr>
<tr>
<td>CMB</td>
<td>0.474</td>
<td>40.0</td>
<td>57.3</td>
<td>61.3</td>
</tr>
<tr>
<td>CNJ</td>
<td>0.545</td>
<td>43.3</td>
<td>68.3</td>
<td>72.3</td>
</tr>
</tbody>
</table>

CMB and CNJ are the methods proposed in Kato, Ohshima, Oyama and Tanak’s paper [1]. We use MRR as the metric to evaluate the experiment result. Our method gets a MRR value of 0.773 which is higher than 0.545. On the percentage of target words at rank 1, our 68 percentage is also much higher than the 43.3 percentage of CNJ method. What’s the most important is that all the target words D is recalled in the top five candidates D.

V. CONCLUSION

In this paper, a Chinese Analogy search method is proposed. We calculate the frequency and weight of word to retrieval the relationship word and target word, this approach takes into account the diversity of the relations between entities. Different relationships between the entities are distinguished by k-means clustering. We test the effect of this method on different type of entity pairs. Our approach achieves a MRR of 0.773 which is higher than existing methods.

ACKNOWLEDGMENT

This work is supported by a grant from the National High Technology Research and Development Program of China (863 Program) (No. 2011AA01A107), the grant from the Shanghai Science and Technology Foundation (No. 11511504000, No. 11511502203, No. 2011XJRHB0078 and No. 10dz2253200).

REFERENCES


