Self-Supervised Synonym Extraction from the Web*

FANGHUAI HU, ZHIQING SHAO AND TONG RUAN
Department of Computer Science and Engineering
East China University of Science and Technology
Shanghai, 200237 P.R. China

Current synonym extraction methods work in a “closed” way. Given the problem word and set of target words, researchers have to choose words synonymous with the problem word using features such as lexical patterns and distributional similarities. This paper tries to discover synonyms in an “open” way and presents a synonym extraction framework based on self-supervised learning. We first analyze the nature of the open method and argue that a trained pattern-independent model for synonym extraction is feasible. We then model the extraction of synonyms from sentences as a sequential labeling problem and automatically generate labeled training samples by using structured knowledge from online encyclopedias and some generic heuristic rules. Finally, we train some Conditional Random Field (CRF) models and use them to extract synonyms from the web. We successfully extract more than 20 million facts, which contain 826,219 distinct pairs of synonyms.

Keywords: synonym extraction, self-supervised learning, sequential labeling, pattern, encyclopedia

1. INTRODUCTION

Synonyms are words with similar or identical meanings. The synonym relation is a typical lexical semantic relation, which is included in most lexical databases and ontologies. These synonym relations are useful in a number of NLP and text mining applications, such as: information retrieval, question-answering, text summarization, language generation and recommendation.

Traditional dictionaries contain a number of synonyms but these dictionaries are written for human searching and are hardly readable by machines. To obtain machine-readable synonym databases, the intuitive method is manual construction. A most successful example is WordNet [1], which contains 117,000 synsets in total, in which the main relation among the words is synonymy. For manually built synonym databases, the accuracy is good but the limitations are obvious and the coverage is too narrow; moreover, they require a large amount of manual work and age quickly.

Therefore, a great deal of research effort has been devoted to automatically extracting synonyms from text [2, 3], dictionaries [4, 5, 6], Wikipedia [7, 8, 9, 10], search engines [11, 12, 13] and so on. According to Turney’s definition [14], the task of recognizing synonyms is, “given a problem word and a set of alternative words, choose the set of alternative words that is most similar in meaning to the problem word”. Most of the current studies work around this definition and thus, work in a closed way. Given both the problem words and the target words, these methods can only determine synonymous relationships amongst them and cannot actively discover new synonymous words.

* This research is supported by the National Science and Technology Pillar Program of China under grant number 2013BAH11F03 and National Nature Science Foundation of China under grant number 61003126.
This paper focuses on extracting synonyms from the web in an open way, that is, both the problem words and the alternative words are unknown. The open method aims to extract as many synonyms as possible; we have to select candidate entities first and then have to derive synonyms from them. Obviously, there are three difficult aspects: how to select candidate entities from the millions of words or entities of a language, how to model the synonym extraction problem from unstructured free text, and how to indicate synonyms among the candidate entities without any prior knowledge (rules) or labeled training samples. The intention of the paper is to find solutions for open synonym extraction. We analyze the feasibility of the open methods, model the synonym extraction problem, generate labeled training samples automatically, train synonym extraction models, extract synonyms and evaluate the results.

The following contributions are made in the article:

- Try to extract synonyms in an open way, analyze the nature of the synonym relation in sentences and prove that open methods are feasible;
- Automatically select the candidate synonym entities, and automatically label training samples by using structured knowledge in Wikipedia and some general heuristic rules;
- Train synonym extraction models and extract synonyms from the web; 826,219 distinct pairs of synonyms are successfully extracted from 2 billion sentences.

The paper is organized as follows. The next section compares the two types of synonym extraction methods, analyses the feasibility of open synonym extraction theoretically and proves it by a practical application of synonym extraction in Chinese. Section 3 introduces the extraction process in detail, including labeling of training samples, training of models and extraction of synonyms. We evaluate the experimental results in Section 4 and then briefly review some related works in Section 5. Finally, Section 6 concludes the paper and gives some directions for future work.

2. PROBLEM ANALYSIS

2.1 Closed Synonym Extraction and Open Synonym Extraction

With the explosive growth of data in the web, information extraction tasks are now transferring from traditional limited data to the open big data. Information extraction from big data has the following characteristics: it usually requires handling heterogeneous corpora at web-scale, the number of extractions will be large and the interested extractions are unanticipated; moreover, there could be hardly any pre-defined knowledge (rules) or labeled training corpus.

The traditional closed synonym extraction methods need to know the problem words and the target words first and then work in one of the following ways: manually building lexical-syntactic rules, learning lexical-syntactic rules in a supervised manner by using a labeled training corpus, or identifying synonyms directly using supervised machine learning methods. These methods violate the web-scale information extraction in three aspects: the number of extractions is limited, they can only extract synonyms from a set of

---

1 Our work is published in http://ssco.zhishimofang.com, in the 同义词 (synonym) field of each entity.
pre-determined words and they need the manual building of rules or a training corpus, which is impossible for web-scale data.

In contrast, open synonym extraction methods introduced in this article satisfy all the requirements: they aim to extract as many synonyms as possible from web-scale corpora, they will exploit synonyms actively and they do not need manually built rules or a manually labeled training corpus. A perfect open synonym extraction method will extract all the synonyms of a language. The problems to be addressed are: Are the open methods possible? Are the extracted results satisfactory? The following sub-section will answer the former question and the latter will be addressed in the experimental section.

2.2 The Feasibility of Open Synonym Extraction

Previous work has demonstrated that semantic relations, such as synonym and hypernym, are often expressed using a set of lexical-syntactic patterns. For example, Hearst [19] successfully extracted a number of hypernym-hyponym relations by using specified lexical patterns like “X such as Y”. Patterns can be identified manually, heuristically or (semi-)automatically. Those methods based on manually identified patterns will have good precision but low recall, as it is unpractical for people to discover all the latent patterns. Therefore, many automatic methods [25, 26] are inspired to acquire additional patterns; these methods usually use a bootstrapping process in a supervised manner.

In synonym extraction, many pattern-based methods [3, 4] have been proposed that are able to extract successfully a particular number of synonyms; thus, we believe that synonym relations are also expressed in a set of patterns. We take the Chinese synonym extraction as a practical example to discuss the feasibility of open synonym extraction. In Chinese, we believe the synonym relations are also expressed with some patterns, although there has been little previous work to demonstrate this. To prove our assumption, we studied a number of synonyms in Chinese in order to see whether there are lexical-syntactic patterns that cover most of the studied sample instances. First, we extracted 100 pairs of synonymous words from the redirection pages of Chinese-Wikipedia [27]. Then, for each pair of synonyms, we extracted 10 different sentences that contained the two synonymous words. Finally, we summarized patterns from these 1000 sentences. We cared the patterns in two levels: the word level and the part of speech (POS) level. The derived lexical-syntactic patterns of the two levels are shown in Tables 1 and 2, respectively (in which E\textsubscript{n} stands for an entity). For the word level, we discovered nearly one hundred patterns and some of those that occurred most frequently are listed. For the POS level, six commonly used patterns cover most of our samples.

In our sample sentences, almost every pattern occurs multiple times and expresses more than one pair of synonyms. When the contextual words around one pair of candidate entities matches a pattern that indicates synonyms, then the two entities are probably synonyms. In other words, whether two entities are synonyms is determined by the contextual words around them but not the entities themselves. This suggests that open methods for synonym extraction are feasible and that learning more patterns will obtain more synonyms.
Table 1. Lexical-syntactic patterns of synonyms at word level.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Ratio</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>E1 又叫(also called as) E2</strong></td>
<td>0.132</td>
<td>北京(Beijing) 又叫 北平(Beiping)</td>
</tr>
<tr>
<td><strong>E1 又称(also named as) E2</strong></td>
<td>0.118</td>
<td>荷花(lotus) 又称 水芙蓉(lotus)</td>
</tr>
<tr>
<td><strong>E1 简称(abbreviated as) E2</strong></td>
<td>0.103</td>
<td>上海(Shanghai) 简称 沪(Hu)</td>
</tr>
<tr>
<td><strong>E1 又名(also named as) E2</strong></td>
<td>0.092</td>
<td>荷花(lotus) 又名 荷花(lotus)</td>
</tr>
<tr>
<td><strong>E1 俗称(commonly called as) E2</strong></td>
<td>0.068</td>
<td>计算机(computer) 俗称 电脑(computer)</td>
</tr>
<tr>
<td><strong>E1 原名(also called as) E2</strong></td>
<td>0.061</td>
<td>老舍(Laoshe) 原名 舒庆春(Shu Qingchun)</td>
</tr>
<tr>
<td>E1 是(is) E2 的(de) 同义词(synonym)</td>
<td>0.053</td>
<td>喜欢(like) 是 喜爱(love) 的同义词</td>
</tr>
<tr>
<td>E1 是(is) E2 的(de) 近义词(synonym)</td>
<td>0.041</td>
<td>美丽(beautiful) 是 漂亮(beautiful) 的近义词</td>
</tr>
<tr>
<td>E1 古称(anciently named as) E2</td>
<td>0.036</td>
<td>西安(Xi’an) 古称 长安(Chang’an)</td>
</tr>
<tr>
<td>E1 是(is) E2 的(de) 简称(abbreviation)</td>
<td>0.031</td>
<td>沪(Hu) 是 上海(Shanghai) 的简称</td>
</tr>
</tbody>
</table>

Table 2. Lexical-syntactic patterns of synonyms at POS level.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Ratio</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>E1, Verb E2</strong></td>
<td>0.289</td>
<td>E1 俗称(commonly called as) E2</td>
</tr>
<tr>
<td><strong>E1 Adv. + Verb E2</strong></td>
<td>0.203</td>
<td>E1 又名(also named as) E2</td>
</tr>
<tr>
<td><strong>E1 Verb E2 Partice Noun</strong></td>
<td>0.140</td>
<td>E1 是(is) E2 的(de) 简称(abbreviation)</td>
</tr>
<tr>
<td><strong>E1 Conjunction E2, Verb Noun</strong></td>
<td>0.122</td>
<td>E1 和(and) E2 是(are) 同义词(synonym)</td>
</tr>
<tr>
<td><strong>E1 Partice Noun Verb E2</strong></td>
<td>0.114</td>
<td>E1 的(de) 同义词(synonym) 是(are) E2</td>
</tr>
<tr>
<td><strong>Others</strong></td>
<td>0.132</td>
<td></td>
</tr>
</tbody>
</table>

3. SELF-SUPERVISED SYNONYM EXTRACTION

In this section, we describe the open synonym extraction process in detail. There are mainly four steps: choosing candidate entities for synonyms, labeling training samples, learning the synonym extraction models, and extracting synonyms using the learned models. We model the synonym extraction from text sentences as a sequence-labeling problem and apply conditional random fields to train synonym extraction models.

3.1 Sequence Labeling Problem

In machine learning, sequence labeling involves the algorithmic assignment of a categorical label for each member of a sequence of observed values. Input X is a sequence of observations and output Y represents hidden sequential states that need to be inferred from the observations; all the output y_i form a chain with an edge between each y_i-1 and y_i, which means that they follow the first-order Markov assumption. Commonly used sequence labeling models are the Hidden Markov Model (HMM), Maximum Entropy Markov Model (MEMM) and Conditional Random Field (CRF). CRF is a discriminative undirected probabilistic graphical model that is used to encode known relationships between observations and to construct consistent interpretations. It is often used for labeling sequential data, such as natural language text and its applications include word segmentation, part-of-speech tagging, named entity recognition, relation extraction and so
3.2 Entity Selection

To extract synonym pairs, the first step is selecting candidate entities. When doing synonym extraction in an open way, base noun phrases are good candidates for entities. “Base noun phrases do not contain nested noun phrases, or optional phrase modifiers, such as prepositional phrases [29]”. For each training sentence or target sentence, after doing word segmentation and POS tagging, the words tagged as “NR” (proper noun) and “NN” (other nouns except proper nouns and temporal nouns) are considered as candidates for entities. Later experiments show that two kinds of phrase modifiers are detrimental to the results and therefore, have to be removed. Luckily, in Chinese this is not difficult; we only have to use two simple heuristic rules to remove them:

- The two words sequentially tagged as “VA” (Predicative adjective) and “DEG” (as a genitive marker and an associative marker) before the candidate entities. For example, in the sentence “美丽的 荷花 也 叫 莲花 (The beautiful Hehua (lotus) also called Lianhua)”, “美丽” and “的” are sequentially tagged as “VA” and “DEG”, they are only as modifiers and should be removed.
- All the words except the last one that occur in a sequence in which all words are tagged as “NR” or “NN”. For example, in the sentence “古城 西安 又 叫 长安 (The archaic city Xi’an also named Chang’an), “古城” and “西安” are both tagged as “NR,” the first one should be removed as it is only a modifier of the second one.

The removed parts are all language modifiers without which the structures and the meaning of sentences are not changed.

Intuitively, we have to choose a lightweight NLP tool to perform the word segmentation and POS tagging instead of standard tools, because the target data are web-scale; however, lightweight NLP tools perform less well and thus, may compromise the result of our experiments. Therefore, we have to make a trade-off between the time complexity and the performance. We experiment with both tools and compare the results in the evaluation section. The standard tools used are the Stanford Word Segmenter and the Stanford POS Tagger.

3.3 Modeling

In synonym extraction, the observable variables $X$, are Chinese word sequences and POS tag sequences and the hidden states $Y$, are tags defined by us that identify entities, synonym relations and other words in $X$; we assume $Y$ is satisfied with the first-order Markov assumption about dependencies among the output states. Therefore, we can model the task as a sequence-labeling problem. We adopt the CRF model, and train a CRF model using PocketCRF and name them SE-CRF.

Candidate entities are determined in Section 3.2 and labeled as ENT. Those pairs of adjacent entities (no other entity between them) within a certain distance (e.g., no more
than four words) are candidate pairs for synonyms and the surrounding contextual words of each pair of entities are seen as potential evidence for synonym relations. These contextual words will be assigned the following labels: S_B, the beginning words of the synonym relation; S_C, the center words of the synonym relation; S_E, the end of the synonym relation; and O, words that do not express a synonym relation. Fig. 1 shows two labeled training samples. The first only uses the word level features and identifies the synonym relation “简称 (abbreviate)” of two entities, “上海 (Shanghai)” and “沪 (Hu)”; the words between the other two adjacent pairs of entities are labeled as O because they are not synonyms. The second sample uses both word level features and POS level features.

![Image 1](image1.png)

(1) 上海简称沪，是中国最大的城市
(Hu is short for Shanghai, is the largest city of China)

![Image 2](image2.png)

(2) 荷花也叫莲花，原产于中国
(Hehua (lotus) also called Lianhua, originally grows in China)

**3.4 Feature Selection**

In the article, the features at word level and POS level are adopted and we do not use any deep-NLP features. On the one hand, we argue that the two-level features are competent enough to train models for synonym extraction; on the other hand, the performance of Chinese deep NLP tools is doubtful, with most achieving a precision under 0.75. Similarly, Jiang and Zhai [28] have demonstrated that for relation extraction, basic unit features were sufficient to achieve satisfactory performance and that the over-inclusion of complex features can compromise the performance.

We use the unigram word (or POS) features and bigram word (or POS) features. For example, a group of word level features we use is “W_2”, “W_1”, “W_0”, “W_1”, “W_2”, “W_1W_0” and “W_0W_1”, where W stands for a word, index 0 indicates the current word in focus and indices -n/n indicate the nth word to the left/right of the current word. Similarly,
an example group of POS level features is “POS_{-2}”, “POS_{-1}”, “POS_{0}”, “POS_{1}”, “POS_{2}”, “POS_{1}POS_{0}” and “POS_{0}POS_{1}”.

Intuitively, the word level features will lead to good precision and the POS level features will raise the recall. In order to measure the importance of the two levels of features, we will train models that use them respectively and together. Our guessing will be verified in the evaluation section. Open information extraction systems usually have to trade precision against recall [29]; by changing features or other factors, we can make a trade between precision and recall.

3.5 Self-Supervised Training

The most important advantage of SE-CRF is that it generates labeled training samples automatically, that is, they are self-supervised. We use the structured data that can be extracted directly from Wikipedia and some generic pattern-independent heuristic rules to generate the training data. As the essence of our self-supervised methods is learning as many as possible patterns that indicate pairs of entities are synonyms, the heuristic rules we used here should be independent to patterns; therefore, we say the rules are “generic pattern-independent”.

The candidate sentences for the training data are sentences that contain at least one pair of candidate entities and the pair of entities must occur within a certain distance. The positive sample sentence is determined by only one simple heuristic rule, “the pair of candidate entities is synonyms”.

However, the heuristic rules for generating negative samples are more complicated and we have to leverage the structured information in Wikipedia in order to determine the pair of entities that are not synonyms. For entity pairs that are not synonyms, we use six heuristic rules in total, two of which are: the two entities are of different types in Wikipedia, for example, one corresponds to a category label and the other is an instance page of the category; the two entities are of different domain in the category system of Wikipedia, for example, one is a plant and the other is an animal. The two samples in Fig. 1, present three pairs of adjacent entities that are not considered synonyms. For the first pair of entities, “沪 (Hu)” and “中国 (China)”, the latter is a category label of the former; and for “莲花 (Lianhua)” and “中国 (China)”, the first is a plant and the second is a country, which are different domains.

The automatically labeled samples are formatted as required and then the CRF learner takes them as input to train the synonym extraction models. The training process is time-consuming and it will usually continue for several days.

3.5 Synonym Extraction

For other input sentences that satisfy the requirement of candidate sentences for training data, by initially performing word segmentation and POS tagging, those candidate entities are recognized by the method described in Section 3.2. Then, SE-CRF makes a single pass over them and labels the contextual words between each candidate pair of entities. If the contextual words of one pair of adjacent entities are labeled with synonym tags, then the two entities will probably be selected as synonyms. We also record in detail how many sentences and how many distinct patterns support every pair of synonymous
words; this supporting information is important in making the trade-off between precision and recall.

4. EVALUATION

To evaluate the performance of SE-CRF, we apply it to more than 20 million web pages that contain about 2 billion sentences. These web pages are extracted from four online encyclopedias (Chinese-Wikipedia, Hudong-Baike\(^6\), Baidu-Baike\(^7\) and Soso-Baike\(^8\)) and some other good domain websites. The reasons for choosing them are that: each contained web page comprises about one hundred sentences on average; the contained web pages contain good information, as they are all written to describe something; they are open and are easy to extract; and the domain websites provide more abundant and deeper knowledge.

The candidate sentences are those with entity pairs within a certain distance, under this condition, we obtain more than 700 million candidate sentences. In order to generate more candidate sentences and obtain more results, we use a heuristic rule to rewrite sentences with caesura signs. In Chinese, caesura signs are used as a splitter of similar paratactic things and thus, removing part of these paratactic things will not break the original sentence. For example, the sentence “荷花又称莲花、芙蓉、菡萏、芙蕖 (Hehua (lotus) also called as Lianhua, Furong, Handan and Fuqu)” can be rewritten into four sentences, “荷花又称莲花 (Hehua (lotus) also called as Lianhua)”, “荷花又称芙蓉 (Hehua (lotus) also called as Furong)”, “荷花又称菡萏 (Hehua (lotus) also called as Handan)” and “荷花又称芙蕖 (Hehua (lotus) also called as Fuqu)”. By the simple rewriting process, we obtain other 90 million candidate sentences; the effects of them will be discussed in Section 4.4. SE-CRF will make a single pass over these sentences and determine whether there are any synonym pairs.

We will evaluate SE-CRF in three aspects: precision, recall and scale, all of which will be specified in Section 4.1. The following three factors will obviously affect our results:

- The features selected for training models. In the article, there are word level features and POS level features. Experienced machine learning researchers may understand that the adoption of appropriate features often matters more than any other factor.
- The distance between each pair of entities. Intuitively, a distance that is too small will generate results with good precision but will omit many synonym pairs that occur at larger distances; distances too large will compromise both the precision and the total number of extractions.
- The threshold (marked as \(th\) in the following sections). This determines one pair of entities as synonyms; specifically, how many sentences are sufficient to support the synonym relation between the two entities.

4.1 Evaluation Metrics

Precision and recall are commonly used metrics in information extraction and there-

\(^6\) http://www.hudong.com/
\(^7\) http://baike.baidu.com/
\(^8\) http://baike.soso.com/
fore, we adopt them to evaluate our system. The precision at a specific threshold \( th \) is defined as the number of correct extractions divided by the total number of extractions at or above \( th \). The recall at a threshold \( th \) is the number of correct extractions at or above \( th \) divided by the total number of correct extractions at all thresholds. Note that the recall is with respect to all the correct results that our system extracted but not all the potentially unknown correct results in the total corpus or the whole web. This is consistent with the recall metric used in TREC: only count correct instances that are in the data collection actually processed by a system.

Scale is used to measure the activeness or open degree of an information extraction system when working in an open way. Usually, the scale is measured in two ways: counting the absolute quantity of the correct extractions, or comparing the number of correct extractions with other start-of-the-art systems.

As with most other information extraction systems, our system has to make a trade-off between precision and recall. As the scale of our system is also very important, we also have to trade precision and recall against scale. By raising the threshold that deems a synonym pair true, we increase precision and decrease recall, whereas lowering the threshold has the opposite effect. However, lowering the threshold will also increase the number of correct extractions.

It is unreasonable to evaluate all the results extracted and thus, we randomly select one thousand sample synonym pairs from the result set. In order to maintain fairness, the sample extractions should be neither with too many nor with too few support sentences, and they should have different support sentences and support patterns. In order to simplify the task of counting the correct extractions, we compare them with a synonym base called “Tongyici-Cilin (extended version)”. If they are in the synonym base, we consider them true; otherwise, we have to evaluate them manually. Every extraction that needs to be evaluated manually is evaluated by experts of different domains; we count the result and determine an extraction as true when more than 80% of experts deem it true.

4.2 Feature and Distance Evaluation

Two types of features are used: the word level features and the POS level features and we train models using them alone and together. In order to eliminate the effect of the distance threshold, we train models with a distance value from 1 to 5 because the five distance values cover more than 95% of the sample sentences that we manually counted in Section 2. The results of precision against recall are shown in Fig. 2 and the total number of correct extractions is shown in Fig. 3.

Fig. 2 implies that word level features play a key role in the synonym extraction process, as in each group the model with only word level features has much better precision. By contrast, adding the POS level features will obviously compromise the precision. Fig. 3 implies that the total number of correct extractions is almost the same whichever kind of features is adopted. Thus, we make the conclusion that the POS level features are not suitable for synonym extraction tasks, whereas word level features perform well.
Fig. 2. The precision-recall results when different features and gaps are chosen. The abscissa denotes recall and the ordinate denotes precision.

Fig. 3. The number of correct extractions when different distances between entity pairs are applied.

We collected the wrong extractions that only occur when POS level features are used and analyzed them carefully. We found that most POS level patterns that may express synonyms can also express other relations, such as hyponyms and therefore, the precision is bad. We also examine the wrong extractions in all cases in order to find causes in addition to the features and conclude that the improper pre-processing, including word segmentation, POS tagging and entity selection, is the main cause for the wrong extractions in all cases. Of course, some wrongly written sentences will certainly lead to incorrect SE-CRF extractions.

Most precision-recall curves are flat until the recall values increase to 0.8, for which
the corresponding threshold value is 3. This means that the synonym relations supported by three or more sentences are believable enough. We find that the wrong extractions that have only one or two supporting sentences can be attributed to the sentences poorly written by a few careless authors.

The precision-recall values gradually decrease when the gap values increase from 1 to 5. However, SE-CRF extracts the largest number of extractions when the distance is 3. When the distance is 1, SE-CRF achieves the best precision-recall values but the total number of extractions is much fewer; this result accords with our conjectures at the beginning of this section. When trading precision and recall against the total number of extractions, we determine that SE-CRF performs best with a distance value of 3, with which the precision-recall values are good and the total number of correct extractions is the greatest.

Table 3. The experimental results when different NLP tools are used for pre-processing.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Precision</th>
<th>Recall</th>
<th>Correct extractions</th>
<th>Total time spent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>0.907</td>
<td>0.8</td>
<td>18,982,415</td>
<td>237 hours</td>
</tr>
<tr>
<td>Lightweight</td>
<td>0.903</td>
<td>0.8</td>
<td>18,773,292</td>
<td>45 hours</td>
</tr>
</tbody>
</table>

4.3 Time Complexity vs. Performance

Section 3.2 mentioned that two types of NLP tools are used: a lightweight one and a standard one. All the former experimental results are based on the lightweight tool. The standard NLP tool requires eight days to pre-process the two billion sentences, whereas a little more than three hours are spent by the same machine using the lightweight tool to perform the pre-processing task. The time spent in extracting synonyms from the pre-processed sentences is almost the same, because the standard tool only obtains a few more candidate sentences. Approximately 800 million candidate sentences are scanned by SE-CRF in about 43 hours. Table 3 shows the results when the two models work at their optimum level, i.e., with a threshold value of 3, distance value of 3 and only word level features used. It is a surprise that the precision values are almost the same. After analyzing some sentences wrongly segmented by the lightweight tool, we find that SE-CRF discriminates correctly as long as the segmentation results are the same in the training samples and testing samples. As the words tend to be considered as candidate entities hardly have any other POS tags, the candidate entity pairs and candidate sentences obtained by the two tools are almost the same; therefore, the total number of correct extractions is almost the same too.

4.4 Global Statistics

When working with the best features of determination threshold value and distance value, we extract 18,773,292 correct facts from the 2 billion sentences with a precision of 0.903. There are a lot of repeated extractions, including synonym pairs with the same support sentences and different sentences; luckily, they are easy to merge as we only have to merge the extractions with the same pair of entities. Finally, we obtain 826,219 pairs of synonyms. Moreover, as the synonym relation is transitive, we merge the extracted synonyms further and obtain 119,374 groups of synonyms.

10 Our machines are Dell R710 servers, which have 32G RAM and each machine has two CPUs with 2.4G main frequency and 16 threads.
In order to find whether there are any other effects (either positive or negative) of the sentence rewriting process mentioned at the beginning of Section 4, we also run the best performed SE-CRF on the candidate sentence set without the 90 million ones generated by the rewriting process. We totally obtain 16,925,923 correct extractions and 802,923 pairs of synonyms, and the precision and the recall are almost not changed. The reduction of the total number of correct extractions is reasonable as the candidate sentences reduced, and their reduction ratios are approximate; while the reduction ratio of the number of synonym pairs is not so great because the synonym relations may also occurred in other sentences.

4.5 Comparison with Other Synonym Bases

We also compare our results with the most famous (and probably the biggest at present) Chinese synonym database named Tongyici-Cilin (Cilin). There are two types of relations in Cilin, one type is called “相等或同义 (equal or synonymous)”, words with this relation are really synonyms; the other is named “同类但不相等 (the same type but unequal)”, words with this relation are not synonyms. Cilin has 77,343 words in total, in which 55,844 ones are organized by the synonym relation into 9,995 groups.

The comparison is shown in table 4, we can conclude that: (1) our synonym base has about nine times more words and twelve times more synonym groups than Cilin; (2) only 13,923 words are in both of the two bases, about a quarter of Cilin. We have compared the two synonym bases carefully and have found that there are mainly five types of words are not (or rarely) collected by Cilin, as shown in Table 5. The reason for (2) is simple: Cilin contains synonyms of many POSes such as noun, verb, adjective, and so on; but our SE-CRF only takes words tagged as “NR” and “NN” as candidate entities; therefore, our synonym base can be considered as an extension of Cilin but not a substitute.

<table>
<thead>
<tr>
<th>Table 4. Comparison with Cilin, only words organized as synonyms in Cilin are listed.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE-CRF</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Words</td>
</tr>
<tr>
<td>Synonym Groups</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5. The synonyms not or rarely collected by Cilin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Names</td>
</tr>
<tr>
<td>孔子 孔丘 仲尼 孔圣人 至圣 至圣先师</td>
</tr>
<tr>
<td>Organizations</td>
</tr>
<tr>
<td>Places</td>
</tr>
<tr>
<td>Chinese-English mapping</td>
</tr>
<tr>
<td>New words</td>
</tr>
</tbody>
</table>

4.6 Discussion

About 300 patterns are learnt by SE-CRF, some of which are really strangely, such
as “$E_1$ 被(bei4) 尊为(respected as) $E_2$”, “$E_1$ 被(bei4) 喻为(likened as) $E_2$”, and so on.

A few patterns are error-learnt, although they really support some correct extractions, they also support more other incorrect extractions; for example, the pattern “$E_1$ ( $E_2$ )” (the body of the pattern is a pair of Chinese round brackets) supports the correct extraction “电脑(computer) ( 计算机(computer) )”\(^{11}\), but it also support incorrect ones like “华盛顿(Washington) ( 人名(name) )” and “华盛顿(Washington) ( 地名(place name) )”. The reason for the learning of this kind of patterns is: they occur in the positive training examples, and there are no (or not enough) negative training examples to specify that they are inappropriate (or not only) for expressing synonym relations.

Most of the learnt patterns are with good reliability; however, almost none of them are absolutely correct, even those most frequently occurred patterns. For example, the sentence “香港人(The Hong Kong People) 又叫(also call) 大洋洲(Oceania) 为(as) 澳洲(Australia)” is supported by the pattern “$E_1$ 又叫(also called as) $E_2$”, however, “香港人” and “大洋洲” are not synonyms.

Moreover, as with other systems that model extraction from natural language text as a sequence-labeling problem, SE-CRF has several limitations; for example, it cannot exploit document-level features and the two target entities must occur in the same sentence within a certain distance. There are two ways to address these disadvantages: the first is to use as large a corpus as possible and the second is to use models that can adopt document-level features.

5. RELATED WORK

Many approaches have been proposed to extract synonyms, including distributional similarity, lexical patterns and Wikipedia and search engine-based methods.

**Distributional similarity-based methods.** Most distributional similarity methods are based on the distributional hypothesis first proposed by Harris [15]. The hypothesis assumes that words with similar meanings tend to appear in similar contexts. Hagiwara et al. [16] illustrated experimentally the importance of contextual information selection for automatic synonym acquisition. They extracted three kinds of word relationships from corpora, including dependency, sentence co-occurrence and proximity. Their experiments showed that dependency and proximity performed well but that a combination of contextual information gave better performance. They also used distributional features as features of a supervised learning-based method [17]. Lin [18] defined a word similarity measure based on the distributional pattern of words and used the similarity measure to construct a thesaurus.

**Lexical pattern-based methods.** Hearst [19] is a pioneer of using lexical patterns to extract word semantic relationships. He applied patterns like “X such as Y” to detect hypernym-hyponym relationships. Earlier researchers mostly used a manual pattern definition approach for information extraction; however, the work is time consuming, requires linguistic skills and it is impossible to find all patterns. Therefore, pattern learning is achieved by supervised approaches, some of which were summarized in [20]. Wang et al. [21] proposed an automatic pattern construction approach by using some seed synonyms (antonyms), which were extracted from WordNet by using heuristic rules. They extracted synonyms and antonyms by combining these patterns to maximize the recall. Simanovsky

\(^{11}\)In order to distinguish from the pattern’s body, we mark the English explanations as subscripts.
Wikipedia-based methods. Wikipedia contains abundant (semi-)structured knowledge, which is good for synonym extraction. Milne et al. [8] successfully constructed a domain-specific thesaurus from Wikipedia. The extracted semantic relations included synonyms, polysemy, taxonomy, etc., and they proved that their thesaurus had high coverage in the corresponding domain. Weale et al. [10] presented an approach to detect synonyms from the graph structure of Wiktionary. Their semantic relation metrics are based on a direct measure of information flow in the graph and a comparison of the list of vertices found to be close to a given vertex.

Search engine-based methods. With the popularity of search engines, the accumulated search log becomes another important corpus for mining information. These methods [12, 13, 22] mainly contain three steps: identify webpage URLs that are strongly related to the entity; identify queries that have clicked these URLs, which are candidate synonym strings of the entity; identify the candidates by using similarity functions named “click similarity”. Chakrabarti et al. [11] summarized a set of simple and natural properties that entity synonyms should satisfy and developed a synonym discovery framework that combined the individual similarity values to produce synonyms.

Some hybrid methods were also proposed to compute word similarities. Mirkin et al. [23] integrated the pattern-based methods and distributional similarity approaches to acquire lexical entailment relationships. Akermi and Faiz [24] used a measure based on an online dictionary and a metric based on page counts returned by a social website.

The limitations of these methods are that they work in a closed way.

6. CONCLUSIONS AND FUTURE WORK

This paper discusses synonym extraction from the web in an open way. We do not have the problem word and target words before extracting but select base noun phrases as candidate entities for synonyms. First, we draw the conclusion that extracting synonyms in an open way is possible by analyzing how synonyms are usually expressed in sentences. Then, instead of manually labeling the training samples, we do the labeling work by using the structured information in Chinese-Wikipedia, together with some general heuristic rules. After that, we train sequence-labeling models with CRF and make a single pass over the candidate input sentences to extract synonyms. Finally, we evaluate the experimental results by comparing them against a Chinese synonym base. We evaluate the precision, recall and scale of the proposed approach and make the conclusion that it has good precision, good recall and large scale. Moreover, when working with the lightweight NLP tool, the time complexity is low but the performance is quite good.

In future, we will solve the problem that all sequence-labeling algorithms suffer by applying models that can use document-level features. We are also considering transferring the methods to the learning of other semantic relationships (such as hypernym).
REFERENCES

23. S. Mirkin, I. Dagan and M. Geffet, “Integrating pattern-based and distributional sim-

**Fanghuai Hu** (胡芳槐) was born in Hunan, China in 1986. He received the B.S. and M.S. degrees in East China University of Science and Technology, Shanghai, China, in 2008 and 2010, respectively. He is currently pursuing a Ph.D. degree in Computer Science and Technology from East China University of Science and Technology. His research interests include information extraction, machine learning and big data analysis.

**Zhiqing Shao** (邵志清) received the Ph.D. degree in Computer Software from Shanghai Jiaotong University in 1998. He is now a Professor in Department of Computer Science and Technology, East China University of Science and Technology, Shanghai, China. He has published more than 100 papers in international conferences and journals. His research interest includes cloud computing, distributed parallel computing, software engineering.

**Tong Ruan** (阮彤) received the Ph.D. degree in Chinese Academy of Science in 2002, Beijing, China. She is now a Professor in Department of Computer Science and Technology, East China University of Science and Technology, Shanghai, China. Her current research interests include big data analysis, software engineering.
Thanks very much to all the reviewers and editors. We are sorry for the belated submission of our revised manuscript as we only pay close attention to the corresponding email but not the journal's submission system.

The following modifications according to the reviewers’ comments are made in our revised version:

<table>
<thead>
<tr>
<th>Comments and response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Review1</strong></td>
</tr>
<tr>
<td>Interesting paper; it is clearly written, the proposed method well explained.</td>
</tr>
<tr>
<td><strong>Response:</strong> none.</td>
</tr>
<tr>
<td>1. This paper proposed a Chinese synonym extraction framework from the web based on self-supervised learning.</td>
</tr>
<tr>
<td><strong>Response:</strong> none.</td>
</tr>
<tr>
<td>2. In Table 1, to understand the distribution of the extracted lexical-syntactic patterns of synonyms, what are their ratios, respectively?</td>
</tr>
<tr>
<td><strong>Response:</strong> The ratios are added into Table 1.</td>
</tr>
<tr>
<td>3. To understand the entity selection of Section 3.2, some examples for removing words are needed. Moreover, besides the presented two rules, how many heuristic rules do you use and how do you decide them?</td>
</tr>
<tr>
<td><strong>Response:</strong> Examples are added in Section 3.2. We only use two rules here in total, as we only find the two main kinds of detrimental phrase modifiers.</td>
</tr>
<tr>
<td>4. In Section 3.4, it is better to indicate the word level features as “W-2”, “W-1”, “W0”, “W1”, “W2”, “W-1,W0”, “W0,W1”. Likewise, “POS-2”, “POS-1”, “POS0”, “POS1”, “POS2”, “POS-1,POS0” and “POS0,POS1”, respectively.</td>
</tr>
<tr>
<td><strong>Response:</strong> Yes, we indeed use the features as you listed above in our experiments, here is just a slip of the pen, we are sorry.</td>
</tr>
<tr>
<td>5. In Section 3.5, what are the generic pattern-independent heuristic rules? Furthermore, how many rules are there? I maybe miss something.</td>
</tr>
<tr>
<td><strong>Response:</strong> We have added the explanations of “pattern-independent heuristic rules” in the first paragraph of Section 3.5, that is, “as the essence of our self-supervised methods is learning as many as possible patterns that indicate pairs of entities are synonyms, the heuristic rules we used here should be independent to patterns; therefore, we say the rules are ‘generic pattern-independent’”. We only used one rule for generating positive examples; for generating negative examples, we applied six rules in total. We have specified them in Section 3.5.</td>
</tr>
<tr>
<td>6. In Section 4, if you did not rewrite the sentences using some heuristic rules, are the evaluation results still the same? How many heuristic rules are there?</td>
</tr>
<tr>
<td><strong>Response:</strong> The evaluation results are a bit different: first, about 90 million more candidate sentences are obtained by the sentence rewriting process; and we have done the experiment and added the discussion of the effect of sentence rewriting in the second paragraph of Section 4.4. Only one rule is applied here.</td>
</tr>
<tr>
<td>Question</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>7. In Section 4, what is the value of threshold ( th ) (support number) in the experiment?</td>
</tr>
<tr>
<td>8. To the best of my knowledge, there is another high quality Chinese ontology named “HowNet”. Why don’t you compare your experimental results with that dictionary?</td>
</tr>
<tr>
<td>9. As to Table 4, for comparing the results with Tongyici-Cilin more discussion is needed, such as coverage etc.</td>
</tr>
<tr>
<td>10. There are two references (16, 22) which are not cited in the paper.</td>
</tr>
<tr>
<td>1. This paper proposes a method to find synonym relations by patterns learned from a training set automatically created from online encyclopedias. By using only lexical surface features, the system achieved 90% and 80% in precision and recall respectively.</td>
</tr>
<tr>
<td>2. Although the authors claimed that the system performed very well, it would be good if the authors can provide some examples of newly-extracted synonyms or synonym-relation patterns for the readers to get a sense how better this system can do than a baseline system using well-known synonym patterns.</td>
</tr>
<tr>
<td>3. It is incredible that the proposed curious system found nearly 7 times of synonym groups than Tongyici-Cilin. What kinds of synonyms were not collected in Cilin? How did the authors prove that these groups are correct?</td>
</tr>
<tr>
<td>4. For error analysis, it would also be great to give some error examples and the correctness of the learned patterns.</td>
</tr>
</tbody>
</table>

Thanks to the reviewers and the editors again.