American Journal of Information Systems, 2017, Vol. 5, No. 1, 27-32
Available online at http://pubs.sciepub.com/ajis/5/1/4

DOI:10.12691/ajis-5-1-4

DACE: Extracting and Exploring Large Scale Chinese Web Collocations with Distributed Computing

Lan Huang*, Juan Zhou, Jing Xue, Yongxing Li, Youfu Du

College of Computer Science, Yangtze University, Jingzhou, Hubei, China
*Corresponding author: lanhuang@yangtzeu.edu.cn

Abstract Words that often occur together form collocations. Collocations are important language components and have been used to facilitate many natural language processing tasks, including natural language generation, machine translation, information retrieval, sentiment analysis and language learning. Meanwhile, collocations are difficult to capture, especially for second language learners; and new collocations develop quickly nowadays, especially with the help of the affluent user generated content on the Web. In this paper we present an automatic collocation extraction and exploration system for the Chinese language: the DACE system. We identify collocations using three measures: frequency, mutual information and χ^2-test. The system was built upon distributed computing frameworks so as to efficiently process large scale corpora. Empirical evaluation and analysis of the system showed the effectiveness of the collocation measures and the efficiency of the distributed computing processes.

Keywords: *information extraction, collocation, MapReduce, Chinese natural language processing*

Cite This Article: Lan Huang, Juan Zhou, Jing Xue, Yongxing Li, and Youfu Du, "DACE: Extracting and Exploring Large Scale Chinese Web Collocations with Distributed Computing." *American Journal of Information Systems*, vol. 5, no. 1 (2017): 27-32. doi: 10.12691/ajis-5-1-4.

1. Introduction

Collocations, also known as multi-word expressions [1] or compound words [2], are important language units. From the computational point of view, a collocation is a set of words that occur together more often than by chance [3]. This include, for example, compound nouns like *train station*, phrasal verbs like *follow up*, proper nouns like *New Zealand*, and common syntactic patterns like adjective+noun and *heavy rain*. Collocations have been used widely in many natural language processing tasks, for example to help natural language generation [4], improve machine translation quality [1], impact search result ranking [5], disambiguate word senses [6] and assist sentiment analysis [7] and second language learning [8].

Though important, collocations are difficult to capture and learn, especially for second language learners, simply because there are so many of them and their forms are extremely diverse. Previous researchers propose to exploit the enormous web resources to discover collocations for language learning [8]. This motivates us to tap into the Chinese web 5-gram corpus [9] for identifying Chinese collocations. The chosen corpus consists of over 39 billion Chinese phrases, each associated with its number of occurrences across over 800 million web pages. After filtering out phrases that occur less than five thousand times, the remaining phrases still generated 40 million collocation candidates. The sheer amount of data presents new challenge to the conventional standalone extracting process. Therefore, we employed the Hadoop distributed computing platform, parallelizing the extraction process with the MapReduce framework, and storing and retrieving the extracted collocations with the distributed database HBase. We name the system DACE to stand for distributed automatic collocation extraction. Using distributed computing, DACE can complete the extraction process within four hours for all 40 million collocation candidates, and retrieves any collocation within a second.

The rest of this paper is organized as following. Next we review related work on automatic collocation extraction with a specific focus on the Chinese language domain. Section 3 explains DACE's system architecture, and Sections 4 and 5 describe the distributed extraction and indexing phases respectively. Section 6 presents experimental setup and discusses empirical experimental results. Section 7 concludes the study.

2. Related Work

Most automatic collocation extraction methods rely on a measure that can quantify the association strength between words in a phrase, so as to determine the co-occurrence of two or more words is indeed statistically more often than by chance. In general, these measures can be categorized into three types: frequency, information theoretic measures and hypothesis test scores. Early studies used co-occurring frequency to identify collocations [10]. Later, mutual information was commonly employed [11]. By comparing the observed number of co-occurrences with the expected co-occuring frequency assuming that the component words were independent, mutual information recognizes those with a co-occuring probability greater than the expected value as collocations. Similarly, hypothesis

tests discover associated events (i.e. words) by comparing to the *null* hypothesis (i.e. assuming independent events). Such tests include comparing the log-likelihood ratio [12], *t*-test [13] and χ^2-test [8]. Besides, position, span and syntactic rules can also be considered in the association measure [14].

More recent studies have proposed several new ways for collocation extraction. For example, linear regression was applied with features covering 84 collocation rules and three linguistic patterns, to quantify the association strength between words in valid collocations [15]. The bilingual word alignment method commonly used in the machine translation field was adapted to the monolingual scenario to extract collocations that co-occur in similar contexts [16]. Multilingual context and multilingual copora have also received increasing attention in recent studies on automatic collocation extraction [17].

For the Chinese language, previous studies have investigated the distinct properties of Chinese collocations [19] and methods for extracting them [1,2,16,19,20,21,22,23]. Table 1 summarizes the recent studies from three aspects: the corpus in use and its scale in terms of number of characters, the association strength measure employed, and the target application if there is one.

Table 1 shows that news corpus was by far the major resource for most automatic collocation extraction systems. Recently in the English language domain, large-scale web resources and multilingual copora have also been used [8,17]. They provide unprecedented affluent resources for the task. Such resources in the Chinese language are yet to be exploit.

Table 1. Summary of Previous Studies on Chinese Collocation Extraction

Study	Corpus	#chars	Measures	Collocation Length	Additional Info	Applications
Zhang 2000 [2]	News, general documents	75m ~ 650m	Mutual Information Context Dependency	n-gram	NA	Information Retrieval
Lu 2003 [19]	News	11m	Frequency-based; Spread	bigram, n-gram	Linguistic features	NA
Qu 2004 [20]	News	59m	Mutual Information	bigram	Use relative word order ratio to filter collocation candidates	NA
Li 2005 [21]	News	11m	Frequency-based; Spread	bigram, n-gram	Taxonomy-based similarity	NA
Piao 2006 [1]	Unknown	0.7m	Log-likelihood; t-score	n-gram	Stop words filtering POS tagging	Machine Translation
Wang 2007 [22]	News	5m	Maximum Entropy	n-gram	Equal vote	Collocation Recognizer
Liu 2011 [16]	News	28m	Word alignment method	bigram	NA	Information Retrieval

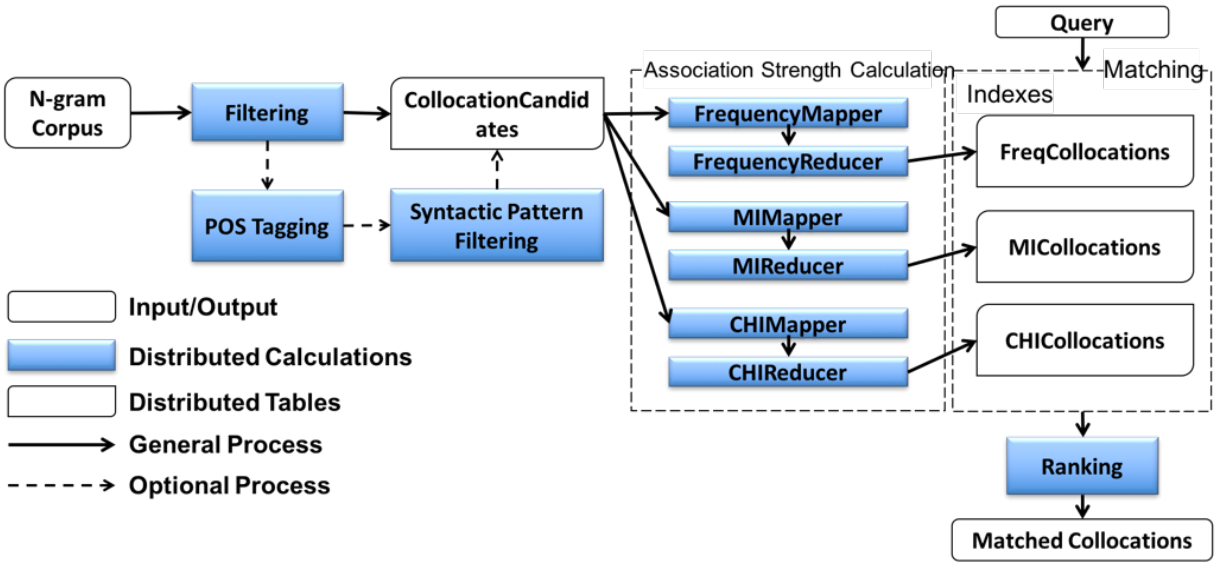

Figure 1. DACE System's Pipeline Architecture

3. System Architecture

The proposed DACE system adopts a pipeline architecture, as shown in Figure 1. Given a corpus consisting of phrases and their number of occurrences, first it will be filtered, for example to remove phrases that contain non-Chinese characters and thus rarely form valid Chinese collocations. Then the phrases can be subjected to syntactic analysis. For example, a part-of-speech (POS) tagger can be employed and then phrases can be filtered based on their syntactic patterns. This step is optional though.

Filtered phrases were then stored into a distributed database table (see Table 2), upon which the association strength of each phrase as a collocation is calculated, by employing different measures. The calculation step is also distributed. The resulting collocations and their associated scores are stored into the index tables.

During exploration stage, i.e. retrieval of the extracted collocations, user usually submit a keyword. It is then matched against the three index tables. Collocations have directions: a collocation can either start or end with the keyword, namely, right and left collocations. DACE provides options to choose the collocation direction, collocation measure, and the number of hits returned. Finally, the matched collocations are ranked in descending order of their associated scores and returned.

4. Distributed Extraction of Collocations

Implementing the DACE system in a distributed environment allows us to perform comparative studies much more efficiently. We selected and implemented three measures for quantifying the salience of a phrase as a collocation: frequency, mutual information and χ^2-test. As for the distributed computing platform, we chose the most well-known off-the-shelf framework Hadoop. This section first explains the collocation measures (i.e. mutual information and χ^2 -test) and then describes the parallelized extraction process.

4.1. Collocation Measures

4.1.1. Mutual Information

Given two words w_1 and w_2, their mutual information is calculated as follow

$$\mathrm{I}(w_1, w_2) = \log_2 \frac{P(w_1, w_2)}{P(w_1) \times P(w_2)} = \log_2 \frac{C(w_1, w_2) \times \mathrm{N}}{C(w_1) \times C(w_2)}, \quad (1)$$

whereas $P(w)$ is the probability of word w, $C(w)$ is w's number of occurrences in a corpus, and N is the total sum of occurrences of all words in the corpus. Mutual information is also known as the pointwise mutual information (PMI).

4.1.2. χ^2-test

χ^2-test is also known as the Pearson's chi-square test. It extracts collocations by comparing the actual and the expected number of occurrences. Given two words w_1 and w_2, their χ^2-test score is calculated as follow

$$\chi^2 = \frac{N \times (c_{11}c_{22} - c_{12}c_{21})^2}{(c_{11} + c_{12}) \times (c_{11} + c_{21}) \times (c_{12} + c_{22}) \times (c_{21} + c_{22})}, \quad (2)$$

where $c_{11} = C(w_1, w_2)$, $c_{12} = C(w_2) - C(w_1, w_2)$, $c_{21} = C(w_1) - C(w_1, w_2)$, $c_{22} = N - c_{11} - c_{12} - c_{21}$, and N is again the total sum of occurrences of all words in the corpus. In words, c_{11} is the number of times w_1 and w_2 co-occur, c_{12} is the number of times w_2 occur without w_1, and c_{21} is the number of times w_1 occur without w_2.

4.2. Parallelized Extraction

We used the Hadoop MapReduce framework to distribute the entire extraction process to a cluster of computing nodes. Each distributed calculation step in Figure 1 corresponds to one Mapper class and one Reducer class in the MapReduce programming framework. In general, the Mapper class processes a phrase and outpts a *<key, value>* pair to represent it, for example, *<collocation, score>*, while the Reducer class mainly sort the collocations by their scores and store them into the backend database. It is worth noting that when calculating mutual information (MI) and χ^2-test (CHI) scores, since they both require summarizing $C(w_1)$, $C(w_2)$ and $C(w_1w_2)$, an intermediate step was designed to collect these statistics (see Section 6.2).

5. Distributed Indexing of Collocations

Efficient indexes are fundamental for responsive retrieval and exploration, especially given large amount of data. In accordance with the extraction process, we used the distributed data storage framework associated with Hadoop—Hbase—as the backend database system.

In contrast to the relational data structure in traditional SQL databases, HBase adopts the column-based data structure. Tables consist of column families, and a column family consists of columns. Both the number and the data type of columns in one column family can vary on the fly. Such dynamic structure is due to HBase's sparse key-value format for physical data storage, as shown in Figure 2. Such structure is perfect for storing the collocation data: the number of collocated words vary dramatically for different key words.

Key				Value
RowKey	ColumnFamily	ColumnID	Timestamp	Value

Figure 2. HBase Sparse Storage Format

Two table structures were designed: one for storing the filtered phrases (i.e. the CollocationCandidates table) and the other for storing the extracted collocations (i.e. the index tables e.g. FreqCollocations). Table 2 and Table 3 illustrate their column-based structures with examples from our dataset.

The CollocationCandidate table consists of two column families, namely *phrases* and *freq*. The *phrases* family records the words in a phrase and their associated POS tag if available, resulting in two (i.e. unigram) to ten (i.e. 5-gram) columns. The *freq* family has only one column that records the number of occurrences of the phrase as a whole. We took the unicode encoding of the phrase as the row key.

Table 2. CollocationCandidate Table Structure with an Example

RowKey	phrases						freq
	term1	pos1	term2	pos2	term3	pos3	
_9e21_86cb_6709 _52a9_4e8e_5065 _5eb7_49	鸡蛋	n	有助于	vp	健康	n	49

Translation of Table 2: Egg (term1); Is good for (term2); Health (term3).

Table 3. Structure of the Index Tables with Examples

RowKey	Collocations											Freq
	term_1	score_1	sp_1	term_2	score_2	sp_2	……	term_n	score_n	sp_n	……	
学习_L	英语	701	n+v	认真	616	adj+v	……	供	259	v+v	……	23964
……												
英语学习_L	在线	23	adj+n+v	免费	10	adj+n+v	……	小学生	5	n+n+v	……	447
……												
学习_R	和	890	n+conj.	方法	321	n+n	……	能力	231	n+n	……	22628
……												
学习方法_R	可以	7	n+n+v	讲座	1	n+n+n	……	总结	1	n+n+n	……	172

Translation of Table 3: Row 1: Learning_L; English (term_1); Hard/Careful (term_2); Provide (term_n).
Row 2: English Learning_L; Online (term_1); Free (term_2); Pupil (term_n).
Row 3: Learning_R; And (term_1); Method (term_2); Capability (term_n).
Row 4: Learning Method; Can (term_1); Lecture (term_2); Summary (term_n).

The three index tables share the same structure, as shown in Table 3. Each word corresponds to two row keys, i.e. two data rows, for phrases start and end with the word respectively. For example, in Table 3, the keyword *学习* corresponds to two rows *学习_L* and *学习_R*, for phrases end (i.e. left collocations) and start (i.e. right collocations) with that key word.

It also has two column families: *collocations* and *freq*. The *collocations* family stores bigrams that either start or end with the word. Phrases are organized with three columns as a set: the following (or preceding) term, collocation score of the phrase (i.e. frequency, mutual information value and χ^2-test value) and its syntactic pattern if available (e.g. *n+v*). Phrases are sorted in descending order of their scores. The last column records the total number of occurrences of the indexing key word in the specified collocation direction. The number of columns in the *collocations* family is huge, and it also varies, for example, in our experiment it varied from zero to 119,942. As exaplained above, HBase's sparse storage format can handle such data efficiently.

6. Results and Discussions

6.1. Experimental Setup

6.1.1. Dataset and Preprocesses

Our dataset is the Chinese web 5-gram corpus, which contains Chinese word n-grams and their observed frequency counts generated from over 800 million tokens of Web text, resulting in over 30G files in gzip format and 39 billion n-grams [9]. The length of the n-grams ranges from unigram (single words) to 5-grams. The corpus is huge. Efficient exploration of such a dataset is challenging.

Considering that non-Chinese characters, such as numbers and English letters or words, rarely occur in real Chinese collocations, phrases that contain these characters were removed from the corpus. We also filtered phrases that occur less then 5000 in the 800 million tokens.

6.1.2. Distributed Computing Platform

We deployed a Hadoop cloud to perform the extraction processes and to support the DACE system. The cloud consists of five computing nodes: two master nodes and three core nodes. Each node was equiped with a 64 bit 16-core CPU with 32G RAM, Huawei's Euler OS 2.2 (an adapted OS based on CentOS), and 40G and 2T SAS disk space for system and data files respectively.

Distributed computing services installed on each node included JDK 1.6, Hadoop 2.7.2 and HBase 1.0.2, and ZooKeeper 3.5.1. The topology structure of the cluster is shown in Figure 3. Node Master1 is the master node and major access point of the cloud.

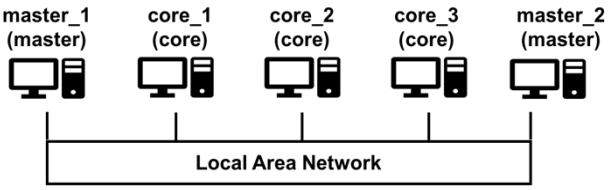

Figure 3. Topology of the Distributed Computing Platform

6.2. Stagewise Analysis

As explained previously, the DACE system mainly consists of two phases: filtering and indexing, as shown in Figure 1. Table 4 compares the data scale and the time cost of each stage.

Table 4. Data Scales and Time Costs in DACE's two major stages

	Filtering	Indexing		
		Frequency	MI	CHI
Data Scale	39 billion	40 million (40,336,116)		
Time Cost (in minutes)	25	24	195	190

Output of the filtering process—the Candidate Collocations table in Figure 1—had 20 million rows. Each row corresponds to a phrase. When broken down into words in the indexing stage, the phrases generated 14 million distinct words, that is, the index tables had 14 million rows. The number of columns in each row varied from zero to 119, 942, resulting in over 40 million distinct expressions. Yet only a small portion of these expressions were valid collocations.

As Table 4 shows, MI and CHI measures took more time than the frequency measure. This is because they involved the three frequency counts: the number of

occurrences of a phrase and of its component words. In practice, we performed a separate step to compute these intermediate scores. This step took about 170 minutes, and since its result was shared by the two measures, the time cost of computing the actual MI and CHI scores took 25 and 20 minutes respectively.

Table 5. Top 20 Collocations Extracted by Different Measures

Frequency								MI		χ^2-test	
No	Collocation	No	Collocation	No	Collocation	No	Collocation	No	Collocation	No	Collocation
1	就 是	11	的 是	21	最 大	31	不 知道	1	耶耶耶耶 耶耶耶耶	1	二二二 二二二
2	不 是	12	都 是	22	一 次	32	我们 的	2	黑鳶 麻鷹	2	肝肝 肝肝
3	我 的	13	的 时候	23	也 不	33	版权 所有	3	手勤 腿勤	3	葔 葔
4	你 的	14	了 一	24	人 的	34	了 我	4	洗盡 鉛華	4	呱呱呱呱 呱呱呱呱
5	的 人	15	不 会	25	是 不	35	她 的	5	汉弗莱 博加特	5	桓桓 桓桓
6	自己 的	16	也 是	26	是 一个	36	有 一	6	酂 醅	6	流流流 流流流
7	中 的	17	他 的	27	好 的	37	并 不	7	仁通 合美	7	素娜 素娜
8	一 种	18	大 的	28	是 我	38	不 到	8	蓼 藍	8	澈澈 澈澈
9	的 一	19	您 的	29	这 是	39	是 在	9	希羅 尤爾	9	王王王 王王王
10	是 一	20	上 的	30	两 个	40	新 的	10	桂林站 柳州站	10	口合 口合

6.3. Comparing Collocation Measures

Table 5 lists the top collocations extracted by different measures. Compared to previous study on the English language [24], similar behavior of the three measures was observed. In general, χ^2-test and mutual information tend to favor expressions with low frequency and has a repetitive pattern. The frequency measure, despite its simplicity, finds meaningful and effective collocations. Therefore, Table 5 lists the top 40 collocations for the frequency measure, and only 10 for the other two measures.

6.4. Collocation Retrieval System

We also implemented a web-based information retrieval system to provide efficient exploration of the extracted collocations, as shown in Figure 4. The search interface provides options to choose the direction of a collocation, the measure, and the number of hits returned. Searching and ranking were also based on HBase queries. We tested ten query times, and the average retrieving time was 258ms.

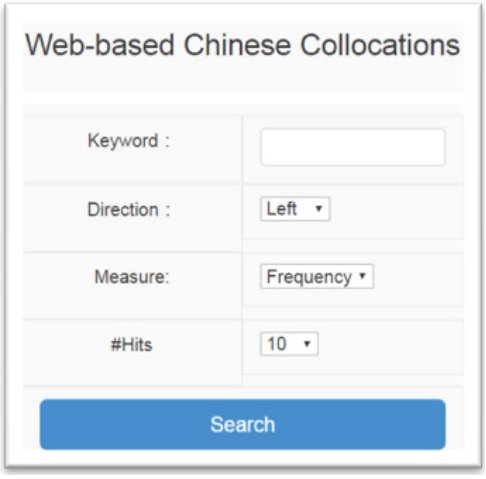

Figure 4. DACE's Search Interface

7. Conclusions

Collocations are important yet difficult to capture. The affluent text on the Web provides natural, updated and valuable resources for automatically extraction of collations. In this paper we designed and implemented the DACE system for automatic collection extraction and exploration. Empiricial experimental results showed that DACE is efficient and the extracted collocations are effective. The search interface of the DACE system is quite simple at the moment, and we plan to improve it with more flexible and user-friendly search options in future.

References

[1] Piao, S. S. L., Sun, G., Rayson, P., Yuan, Q. Automatic Extraction of Chinese Multiword Expressions with a Statistical Tool. In Proceedings of the Workshop on Multi-word-expressions in a Multilingual Context. In Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006), 2006, pp. 17-24.

[2] Zhang, J., Gao, J., Zhou, M. Extraction of Chinese compound words—an experimental study on a very large corpus. In Proc. of the 2nd Chinese Language Processing Workshop, ACL 2000, 2000.

[3] Mckeown, K. R., Radev, D. R. Collocations. In A Handbook of Natural Language Processing, R. Dale, H. Moisl, and H. Somers Eds. Marcel Dekker, New York, 2000, pp. 507-523.

[4] Smadja, F., McKeown, K. Automatically extracting and representing collocations for language generation. In Proceedings of the 28th annual meeting on Association for Computational Linguistics, 1990, pp. 252-259.

[5] Liu, Z. Y., Wang, H., Wu, H., Liu, T., Li, S. Reordering with source language collocations. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 1035-1044.

[6] Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics. 1995, pp. 189-196.

[7] Xu, R. F., Xu, J., Kit C. HITSZ_CITYU: Combine collocation, context words and neighboring sentence sentiment in sentiment adjectives disambiguation. In Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pp. 448-451.

[8] Wu, S. Q., Franken, M., Witten, I. H. Supporting collocation learning with a digital library. Computer Assisted Language Learning, 2010, 23(1), pp. 87-110.

[9] Liu, F., Yang, M., Lin D. Chinese Web 5-gram Version 1 LDC2010T06. Web Download. Philadelphia: Linguistic Data Consortium, 2010, https://catalog.ldc.upenn.edu/LDC2010T06.

[10] Choueka, Y., Klein, S. T., Neuwitz, E. Automatic retrieval of frequent idiomatic and collocational expressions in a large corpus. Journal of the Association for Literary and Linguistic Computing, 1983, 4, pp. 34-38.

[11] Church, K. Hanks, P. Word association norms, mutual information, and lexicography. Journal of Computational Linguistics, 1990, 16, pp. 22-29.

[12] Dunning, T. Accurate methods for the statistics of surprise and coincidence. Journal of Computational Linguistics, 1993, 19, pp. 61-74.

[13] Manning, C., Schütze, H. Foundations of statistical natural language processing. MIT Press. 1999.

[14] Smadja, F. Retrieving collocations from text: Xtract. Computat. Linguist., 1993, 19, pp. 143-177.

[15] Pecina, P.. An Extensive Empricial Study of Collocation Extraction Methods. Proceedings of the ACL Student Research Workshop, 2005, pp. 13-18.

[16] Liu, Z. Y., Wang, H., Wu, H., Li, S. Two-word collocation extraction using monolingual word alignment method. 2011, ACM Transaction on Intelligent Systems and Technology, 3(1), 16.

[17] Seretan, V., Wehrli, E. Multilingual collocation extraction: issues and solutions. In proceedings of the workshop on multilingual language resources and interoperability, 2006, pp. 40–49.

[18] Sun, M. S., Huang, C. N., Fang, J. A Quantitative Analysis of Chinese Collocation. Studies of the Chinese Language, 1997(1), pp. 29–38. (in Chinese)

[19] Lu, Q., Li, Y., Xu, R.. Improving Xtract for Chinese collocation extraction. In Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering. 2003, pp. 333-338.

[20] Qu, W. G., Chen, X. H., Ji, G. L. Automatic Extraction of Word Collocation Based on Frame. Computer Engineering, 2004, 30(23), pp. 22-24. (in Chinese)

[21] Li, W., Lu, Q., Xu, R. Similarity based chinese synonym collocation extraction. International Journal of Computational Linguistics and Chinese Language Processing. 2005, 10, pp. 123-144.

[22] Wang, S. G., Yang, J. L., Zhang, W. Chinese Verbs and Verbs Matching Based on Maximum Entropy Model and Voting Method. Journal of Chinese Computer Systems, 2007, 28(7), pp. 1306-1309. (in Chinese)

[23] Xu, R. F., Lu Q., Wong, K. F., Li, W. J. Building a Chinese collocation bank. International Journal of Computer Processing of Languages, 2009, 22 (1), pp. 21-47.

[24] Wu, S. Q. Supporting Collocation Learning. PhD thesis, 2010.