Using Verb Dependency Matching in a Reading Comprehension System

Kui Xu and Helen Meng

Human-Computer Communications Laboratory,
Department of Systems Engineering & Engineering Management,
The Chinese University of Hong Kong, Hong Kong SAR, China
{kxu, hmmeng}@se.cuhk.edu.hk

Abstract. In this paper, we describe a reading comprehension system. This system can return a sentence in a given document as the answer to a given question. This system applies bag-of-words matching approach as the baseline and combines three technologies to improve the result. These technologies include named entity filtering, pronoun resolution and verb dependency matching. By applying these technologies, our system achieved 40% HumSent accuracy on the Remedia test set. Specifically, verb dependencies applied in our system were not used in previous reading comprehension systems. In addition, we have developed a new bilingual corpus (in English and Chinese) - the ChungHwa corpus. The best result is 68% and 69% HumSent accuracy when the system is evaluated on the ChungHwa English and Chinese corpora respectively.

1 Introduction

Recently, there has been increasing interest in the research of question answering (QA) systems. Researchers in the field of information retrieval (IR) have paid much attention to this topic. One branch in the study of QA system is based on the context of the reading comprehension task. This task was proposed as a method for evaluating Natural Language Understanding (NLU) technologies by a research group at MITRE Corporation [7]. In this task, MITRE Corporation developed the Remedia corpus to evaluate a reading comprehension (RC) system. In addition, another task about question answering on a large-scale is the Text REtrieval Conference Question Answering (TREC QA) track. There is a major difference between TREC QA and reading comprehension. For a given question, TREC QA systems retrieve documents in a large collection of data and then find the answer (in phrases or sentences) within the retrieved documents. Reading comprehension systems only look for answers to given questions within a given story. In this paper, we only address the QA system for reading comprehension task.

Many reading comprehension systems [7], [12], [5] assume that there are common words shared between questions and answers. A reading comprehension system measures the similarity between questions and answers by matching with different features. Features can be as simple as bag-of-words or bag-of-verbs [7],

[12], [5]. In related work such as the TREC QA systems, syntactic and semantic features that have been applied include syntactic parse trees, dependencies and predicate-argument structures [4], [9]. However, syntactic and semantic features are not broadly used for reading comprehension tests.

In this paper, we have developed a reading comprehension system that uses syntactic features in an attempt to improve the accuracy. Syntactic features are represented by verb dependencies in our system. As Allen described in [1], the context-independent meaning of a sentence can be represented by logical forms, which can be captured using relationships between verbs and noun phrases. In this paper, verb dependencies are defined as lexical dependencies in which the heads of dependencies are verbs. We apply verb dependencies to handle ambiguities among candidate sentences when a reading comprehension approach (e.g. BOW matching) cannot discriminate among multiple candidate sentences. We believe that matching with syntactic features like verb dependencies can perform better selection among candidate answers than the simple approach of BOW matching.

In addition, this paper reports our first attempt in developing a Chinese reading comprehension system. We begin by collecting the ChungHwa corpus, which is a bilingual corpus both in English and Chinese.

2 Related Work

In 1999, a group at MITRE developed a reading comprehension system, Deep Read [7]. This system used the bag-of-words (BOW) matching and automated linguistic processing to achieve 36% HumSent[1] accuracy in the Remedia test set [7]. The system applied linguistic processing such as stemming, named entity (NE) recognition, named entity filtering, semantic class identification and pronoun resolution. If multiple candidate sentences contain the maximum number of matching words in BOW matching, the first (earliest occurrence) candidate sentence will be returned as the final answer.

Riloff and Thelen [12] developed a rule-based system called Quarc and achieved 39.7% HumSent accuracy in the Remedia test set. Quarc used not only BOW matching but also a number of heuristic rules that look for lexical and semantic clues in the questions and stories. For example, the WHERE rule,

if contain(S, LOCATION), then Score(S)+=6,

can be interpreted as the following: for *where* questions, if a candidate sentence contains LOCATION, this rule will reward the candidate sentences with 6 points.

[1] By comparing the system answers with the human marked answers, the percentage of correct answers is used as HumSent accuracy. In other words, if the system's answer sentence is identical to the corresponding human marked answer sentence, this question scores one point. Otherwise, the system scores no point. HumSent accuracy is the average score across all questions.

Charniak et al. [5] used bag-of-verbs (BOV) matching, “Qspecific” techniques, named entities, etc. to achieve 41% HumSent accuracy. The BOV matching is a variation of BOW matching in which only verbs are examined instead of all non-stop words. The “Qspecific” techniques use different strategies for different questions. For example, the following is one of the strategies for *why* questions.

> If the first word of the matching sentence is “this”, “that”, “these” or “those”, select the previous sentence.

Ng et al. [11] used a machine-learning approach (C5 learning algorithm) to determine if a candidate sentence is the answer to a question based on 20 features such as “Sentence-contains-Person”, “Sentence-is-Title” etc. This approach achieves 39.3% HumSent accuracy.

As mentioned above, some of the previous work assumed that there is a high degree of overlap between the words used in a question and those used in its correct answer. In our approach, we assume that there is a structural overlap between the syntactic structure of a question and that of the correct answer. Syntactic structures are not commonly used in previous work for reading comprehension tests. In this paper, the impact of applying syntactic structure is examined.

In addition, some rules or features [11], [12] have been applied based on the observation in the training corpus. For example, the dateline (the line that shows the date when the story happened) in a Remedia story can be the answer to a question. Riloff and Thelen [12] applied dateline rules to handle the dateline. For example:

> if contain(Q, story), then Score(DATELINE)+=20

can be interpreted as the following: for *where* and *when* questions, if a question contains “story”, this rule will reward the dateline with 20 points. In addition, Ng et al. [11] used “Sentence-is-Dateline” as a feature in their machine-learning approach. Such corpus-specific technologies do not have impact on the corpus that does not have datelines. In our first attempt, corpus-specific technologies are not involved. We applied a general approach that is corpus-independent.

3 Verb Dependencies

In the current work, verb dependencies are used to represent the syntactic structures of sentences. They can be used as auxiliary information to perform matching based on BOW matching. Verb dependencies can be obtained from parse trees of sentences. In the study of parsing technologies, lexical dependencies have been used to handle ambiguities among parse tree outputs by a PCFGs parser [6]. Research in parsing technologies [6] shows the power of lexical dependencies in improving the performance of a parser. For the same reason, if reading comprehension approaches (e.g. BOW matching) cannot discriminate among multiple candidate sentences, lexical dependencies can also be used to handle ambiguities among candidate sentences.

In the work of Collins [6], the author defines a dependency as a relation between two words in a sentence (a *modifier* and a *head*), written in:

$$< modifier \rightarrow head > .$$

If the head of a dependency is verb, we refer this type of dependency verb dependency. Dependencies can be extracted from syntactic parse trees according to the work of Collins [6]. Fig. 1, Fig. 2 and Fig. 3 show the parse trees of the following question and candidate sentences respectively:

What is the new machine called?
A new machine has been made.
The machine is called a typewriter.

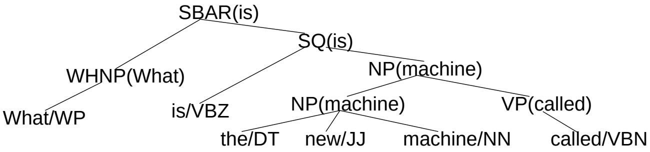

Fig. 1. The parse tree of "*What is the new machine called?*"

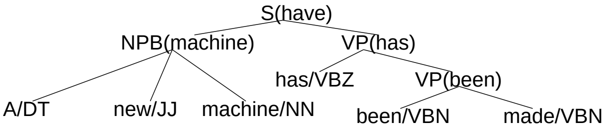

Fig. 2. The parse tree of "*A new machine has been made.*"

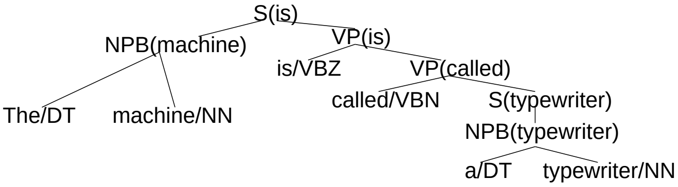

Fig. 3. The parse tree of "*The machine is called a typewriter.*"

These parse trees are lexicalized parse trees since each node in the trees have a label and a headword. The label indicates a syntactic category (e.g. SBAR, NP and VP in Fig. 1). The headword is defined as the prime constituent of a phrase or sentence [2]. For example, the headword of a noun phrase is the noun; that of a verb phrase is the verb. Hence the headword of the noun phrase "a new machine" in Fig. 2 is "machine". Remaining words in the phrase or sentence are regarded as the left or right modifiers of the headword. More specifically, a context-free grammar rule may be represented as:

$$p \rightarrow l_n \ldots l_1 h r_1 \ldots r_m,$$

where p is the parent non-terminal, h is the head child of p, l_i and r_i are the left and right modifiers of h respectively. If h is a non-terminal, the headword of p is the headword of h. If h is a terminal, the headword of p is h. We use the guidelines described by Collins [6] in determining the head constituent in a context-free rule. The procedure of assigning every non-terminal (every non-leave node in parse trees) a headword is called lexicalization [6]. An edge in parse trees involves two nodes: a child node and a parent node. A dependency can be written as:

$$< hc \rightarrow hp >,$$

where *hc* is the headword of the child node, *hp* is the headword of the parent node, *hc* is not identical to *hp*. For the root node in a parse tree, a dependency can be written as:

$$< hr \rightarrow TOP >,$$

where *hr* is the headword of the root node. Only the dependencies whose *hp* or *hr* are verbs are selected as verb dependencies. We call this selection process verb dependency extraction. The corresponding verb dependencies of parse trees in Fig. 1, Fig. 2, and Fig. 3 can be found in Table 1. In this paper, two dependencies match if and only if they are identical.

In BOW matching, if the system returns an incorrect answer, one of the following two cases happens.

- The incorrect answer has a greater number of matching words than the correct answer.
- The incorrect answer and the correct answer have an equal number of matching words but the incorrect answer appears earlier in the document.

Therefore, we can use verb dependencies to distinguish the correct answers from the incorrect answers. For example:

Question: What is the new machine called?
BOW: {be machine new call}

Suppose there are two candidate sentences that have the greatest number of matching words. The matching words in the candidate sentences are displayed in italic in the following:

Candidate sentence 1: A new machine has been made.
BOW: {*be machine new* has make}
Candidate sentence 2: The machine is called a typewriter.
BOW: {*be call machine* typewriter}

Both candidate sentences have the maximum number (three) of matching words among all candidate sentences. The parse trees of the question and two candidate sentences are shown in Fig. 1, Fig. 2, and Fig. 3 respectively. Their corresponding verb dependencies are shown in Table 1.

Table 1. Verb dependencies for the syntactic parse trees in Fig. 1, Fig. 2, and Fig. 3

Verb dependencies for the parse tree in Fig. 1	{<what→be>, <*machine*→*be*>, <*be*→*TOP*>}
Verb dependencies for the parse tree in Fig. 2	{<machine→have>, <be→have>, <make→be>, <have→TOP>}
Verb dependencies for the parse tree in Fig. 3	{<*machine*→*be*>, <call→be>, <typewriter→call>, <*be*→*TOP*>}

The candidate sentence 2 (see Fig. 3) has the maximum number of matching dependencies against the question. Verb dependency matching selects candidate sentence 2 as the final answer, which is the true answer in this example.

4 The Reading Comprehension System

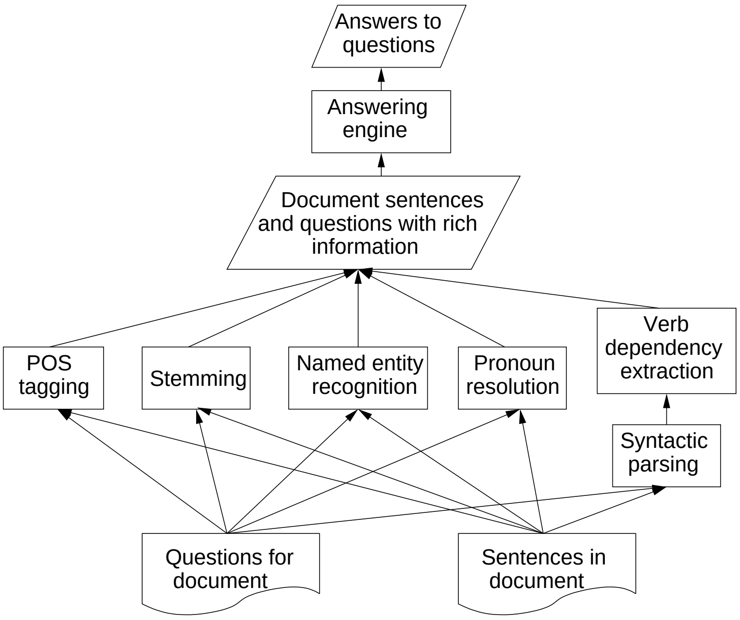

Fig. 4. The flow chart of our reading comprehension system

The flow chart of our reading comprehension system is shown in Fig. 4. In our current system, six processes are applied to identify different types of information in the story sentences and questions. These six processes are part-of-speech (POS) tagging, stemming, named entity recognition (NER), pronoun resolution, syntactic parsing and verb dependency extraction. Three out of the six processes are implemented by the use of natural language processing tools.

- The Brill POS tagger [3] is applied to perform POS tagging.
- A C programming language function (morphstr) provided by WordNet [10] is used to find the base form of nouns and verbs. This process is our stemming process.
- The Collins' parser [6] is applied to obtain the syntactic parse trees of sentences. This is the syntactic parsing process.

In addition, named entity recognition and pronoun resolution are not implemented in current system. We applied the named entity information and the pronoun resolution information that have been annotated in the corpus. These two processes will be studied in our future work. Further more, we followed the method described in Sect. 3 to implement the process of verb dependency extraction.

With these six processes, syntactic and semantic information is obtained to enrich the document sentences and questions. Such enrichment is used by the answering engine to retrieve the answer to a given question.

4.1 Answering Engine

The answering engine applied four technologies to find answers. They are BOW matching, named entity filtering, pronoun resolution and verb dependency matching. We use BOW matching as the baseline. Other three technologies are combined into the baseline incrementally to examine their impact.

BOW Matching. Each sentence in the story is regarded as a word set. BOW matching is conducted between the question word set and the candidate answer word set. Referring [7], a BOW matching system "measures the match by size of the intersection of the two word sets". If multiple candidate sentences contain the maximum number of matching words, the candidate sentence that appeared earlier is returned as the answer. Stop words in the word sets are removed before matching. The stop word list contains 16 words. They are: *the, of, a, an, it, and, or, do, what, where, why, who, how, when, which, all.* In addition, the nouns and verbs in the BOW are replaced by their base forms, which are the outputs of the stemming process.

Named Entity Filtering. Five named entity types (PERSON, ORGANIZATION, TIME, DATE and LOCATION) are used to perform answers filtering for three types of questions (*who, where, when*). The relationships are listed as the following [7]:

- For *who* questions, a candidate sentence that contains PERSON or ORGANIZATION is assigned higher priority.
- For *where* questions, a candidate sentence that contains LOCATION is assigned higher priority.
- For *when* questions, a candidate sentence that contains TIME or DATE is assigned higher priority.

Pronoun Resolution. For pronoun resolution, we replace five pronouns (*he, him, his, she and her*) with their referents in the word sets of candidate sentences and questions for BOW matching. In addition, other pronouns (e.g. *their, them, they, you, your, it*, etc.) and noun phrase referents are also annotated in the corpus. The system also examines the impact when all pronouns and noun phrase are replaced with their referents beside the above mentioned five pronouns.

Verb Dependency Matching. Verb dependencies are extracted according to the process in Sect. 3. Just like words in the word sets of questions and candidate sentences, verb dependencies are inserted into the corresponding word sets for matching.

5 Corpora

5.1 The Remedia Corpus

The Remedia corpus has been used by many researchers in previous work [5], [7], [11], [12]. It is published by Remedia Corporation. MITRE Corporation has annotated named entities, co-reference of pronouns and noun phrase anaphor and the true answer sentences on this corpus [7]. The corpus contains 55 training stories and 60 testing stories. Each story contains 20 sentences on average. There are about 20K words in this data set. For each story, five types of questions are asked: *who, what, when, where* and *why* question. Within the 60 test stories, there are 59 *who* questions, 61 *what* questions, 60 *when* questions, 60 *where* questions and 60 *why* questions. In total, 300 questions are asked in the Remedia test set. In each story, the first line is the title; the second line is the dateline. Other lines are story sentences.

5.2 The ChungHwa Corpus

The ChungHwa corpus comes from a bilingual book, “English Reading Comprehension in 100 days” which is published by Chung Hwa Book Co.,(H.K.) Ltd. This corpus contains 100 reading comprehension stories both in English and Chinese. The following domains are covered in the corpus: the English language, tourism, culture and society, sports, history and geography, arts, literature, economy and business, science and technology. We reserve 50 documents as the training set and the other 50 documents as the test set. The average number of sentences of each document is 9 (varies from 4 to 18). There are about

18K English words and 17K Chinese characters in the ChungHwa corpus. Each document has four questions on average. A linguistic was asked to annotate the named entities, anaphor referents and answer sentences for each document.

Table 2. A sample story from the ChungHwa corpus with an English story and its questions as well as a Chinese story and its questions

Imagine this: you have just won a competition, and the prize is an English language course at a famous school in Britain or the United States. You can either take a 30-week course for four hours a week, or a four-week course for 30 hours a week. Which one should you choose? . . .
If you win a competition, what may be the prize? What may be the two kinds of courses? What is the advantage and disadvantage of the long course?
想像一下：你刚赢得一场比赛，其奖赏是在英国或美国的一所名牌大学学习一门英语语言课程。你可以选一门为30周的课程，每周学习4小时，或者选一门为期4周的课程，每周30小时。你将作何选择？. . .
如果你赢得一场比赛，奖赏或许是什么？ 两类课程会是什么？ 时间长的课程有什么优点和缺点？

Table 2 shows a sample story from the ChungHwa corpus both in English and in Chinese. Within the 50 test stories, there are 16 *who* questions, 98 *what* questions, 7 *when* questions, 17 *where* questions, 11 *why* questions, 10 *yes/no* questions, 10 *how many* questions and 25 *how* questions. In total, 194 questions are asked in the ChungHwa test set.

6 Experimental Results

For English stories, BOW matching, named entity filtering, pronoun resolution and verb dependency matching are applied step by step to examine the impact of each technology. The results are shown in Table 3 and Table 4 for the Remedia corpus and the ChungHwa corpus (the English part) respectively. For the purpose of comparison with previous work, the results in Table 3 and Table 4 are shown with HumSent accuracies. The abbreviations of different technologies are listed below:

- BOW: bag-of-words matching
- PR: the referents of *he, him, his, she* and *her* are resolved
- AA: all anaphors include pronouns in PR, other pronouns (e.g. *their, them, they, you, your, it*, etc.) and noun phrases anaphors are resolved
- NEF: named entity filtering
- VD: verb dependency matching

Table 3. Detail results by using different technologies on the Remedia test set

	BOW	BOW+PR	BOW+PR+NEF	BOW+AA+NEF	BOW+AA+NEF+VD
Who	34%	37%	49%	51%	54%
What	31%	33%	33%	33%	33%
When	33%	37%	60%	62%	62%
Where	28%	32%	28%	30%	30%
Why	15%	18%	18%	22%	23%
Overall	28%	31%	38%	39%	40%

Table 4. Overall results by using different technologies on the ChungHwa test set in English

	BOW	BOW+PR	BOW+PR+NEF	BOW+AA+NEF	BOW+AA+NEF+VD
Overall	67%	68%	68%	68%	68%

The Deep Read system achieved 29% and 36% HumSent accuracy with BOW and BOW+PR+NEF respectively [7]. With the same technologies, our results are comparable to the results of Deep Read system. The difference may caused by different stop words list and stemming process.

As our first attempt on the ChungHwa Chinese stories, BOW, PR and AA have been applied. Currently, a Chinese syntactic parser is not available in our system. So the verb dependencies of Chinese are not used in our experiments. Our system achieved 69%, 69% and 70% HumSent accuracy with BOW, BOW+PR and BOW+AA respectively. Since the Chinese character segmentation has been annotated in the corpus, we simply use the segmentation annotation instead applying a segmentation tool in current study. PR for Chinese only focuses on “他” and “她”. Named entity filtering technology is not used for the Chinese part because we do not have a question classification for Chinese questions. Questions can be more complicated in Chinese to ask time, location and person. Without these three classes, the corresponding named entities cannot be used to perform named entity filtering. We will study question classification and Chinese parsing in our future work.

7 Discussion

In Table 3, our general approach achieved 40% HumSent accuracy, which is comparable to the state of the art, 41% [5]. After applying VD, our system can improve 1%. To further analyze the result, we performed VD alone on the Remedia test set and studied the verb dependency matching situation in the true answer and its question for all 300 questions. We only found that 79 questions have matching verb dependencies against their true answers. That means the accuracy upper-bound is 79/300 = 26.3% when VD is applied alone. Moreover, 74 out of 79 questions can be correctly answered by BOW+AA+NEF. Therefore, the improvement upper-bound using VD is (79-74)/300 = 1.67%. The limited coverage and improvement room of VD lead to the insignificant improvement.

In Table 3, the accuracies increase for *who* and *when* questions after applying NEF. However, the accuracy drops from 32% to 28% for where questions. After manually analyzing the Remedia training set, we found that not all LOCATION tags are recognized. For example:

Question: Where do these sea animals live?
True answer: She was born in a sea animal park called Sea World.
System returned answer: (ORLANDO, FLORIDA, September, 1985) -

In this example, the system returned answer is wrong. “A sea animal park” in the true answer is not tagged as LOCATION. On the other hand, “ORLANDO, FLORIDA” is tagged as LOCATION. Even though the true answer has the maximum number of matching words against the question, the named entity filtering process gave higher priority to the sentence:

(ORLANDO, FLORIDA, September, 1985) -.

The insufficient LOCATION tags in the Remedia corpus lead to the decrease of accuracy of *where* questions.

When manually analyzing the Remedia training set, we found that inference technology and world knowledge are helpful in answering about one third of questions. For example:

Question: Who had a baby at Sea World?
True answer: Baby Shamu’s mother is named Kandu.
System returned answer: The workers at Sea World will watch Baby Shamu each day and make notes.

In this example, the true answer has no matching words against the question. In order to answer the question correctly, the system must know that the one who had a baby is a mother.

For the ChungHwa English stories, the overall accuracies are greater than those of the Remedia corpus. After manually analyzing the ChungHwa training set, we found that questions in ChungHwa corpus tend to use the same words that used in their corresponding answers. That causes the baseline (BOW matching) result is higher than the result obtained from the Remedia corpus. With a higher baseline result, the improvement made by PR, AA, NEF and VD is not obvious.

8 Conclusion

In this paper, we describe a reading comprehension system. This system can return a sentence in a given document as the answer to a given question. This system applies BOW matching approach as the baseline and combines three technologies to improve the result. These technologies include named entity filtering, pronoun resolution and verb dependency matching. By applying these technologies, our system achieved 40% HumSent accuracy on the Remedia test set. Specifically, our system brings in verb dependencies that can be derived from syntactic parses which is not used in previous reading comprehension systems. The verb dependency matching does not lead to significant improvement because of its limited coverage and improvement room. In addition, we have developed a new bilingual corpus, ChungHwa corpus. The evaluation result on the English corpus is 68% HumSent accuracy and on Chinese corpus is 69% HumSent accuracy. In our future work, named entity recognition approaches and pronoun resolution approaches will be studied for English reading comprehension. Moreover, Chinese question classification approaches and Chinese parsing technologies will be studied for Chinese reading comprehension.

References

1. Allen, J.: Natural Language Understanding. The Benjamin/Cummings Publishing Company, Menlo Park, CA (1995)
2. Bloomfield, L.: An Introduction to the Study of Language. Henry Holt and Company, New York (1983)
3. Brill, E.: A Simple Rule-based Part of Speech Tagger. In Proceedings of the Third Conference on Applied Natural Language Processing (1992)
4. Buchholz, S.: Using Grammatical Relations, Answer Frequencies and the World Wide Web for TREC Question Answering. In Proceedings of the tenth Text Retrieval Conference (TREC 10) (2001) 502-509
5. Charniak, E., et al.: Reading Comprehension Programs In a Statistical-Language-Processing Class. In ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems (2000)
6. Collins, M.: Head-Driven Statistical Models for Natural Language Parsing. PhD thesis (1999)
7. Hirschman, L., Light, M., Breck, E. and Burger, J.: Deep Read: A Reading Comprehension System. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (1999)
8. Light, M., Mann, G. S., Riloff, E. and Breck, E.: Analyses for Elucidating Current Question Answering Technology. Journal of Natural Language Engineering (2001) Vol. 7, No. 4
9. Litkowski, K. C.: Question-answering Using Semantic Relation Triples. In Proceedings of the eighth Text Retrieval Conference (TREC 8) (1999) 349-356
10. Miller, G.: WordNet: an On-line lexical database. International Journal of Lexicography (1990)
11. Ng, H. T., Teo, L. H., Kwan, L. P.: A Machine Learning Approach to Answering Questions for Reading Comprehension Tests. In Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (2000)
12. Riloff, E. and Thelen, M.: A Rule-based Question Answering System for Reading Comprehension Test. In ANLP/NAACL-2000 Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems (2000)