AUTOMATIC TEXT ANNOTATION FOR QUESTIONS

Gang Liu, Zhi Lu, Tianyong Hao and Liu Wenyin
Dept. of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, HKSAR, China

Keywords: Text annotation, Similarity, Question answering, Tagger ontology.

Abstract: An automatic annotation method for annotating text with semantic labels is proposed for question answering systems. The approach first extracts the keywords from a given question. Semantic label selection module is then employed to select the semantic labels to tag keywords. In order to distinguish multi-senses and assigns best semantic labels, a Bayesian based method is used by referring to historically annotated questions. If there is no appropriate label, WordNet is then employed to obtain candidate labels by calculating the similarity between each keyword in the question and the concept list in our predefined Tagger Ontology. Experiments on 6 categories show that this annotation method achieves the precision of 76% in average.

1 INTRODUCTION

It has been shown that annotating text with appropriate tags may benefit many applications (Cheng et al., 2005). Such annotated information could provide clues for many information retrieval (IR) tasks to improve their performance, such as question answering, text categorization, topic detection and tracking, etc. In this paper, we address the problem of automatically annotating a special kind of text which is referred to as unstructured questions in question answering (QA) systems.

The past decade has seen increasing research on the usage of QA for providing more precise answers to users' questions. As a consequence, there are some automatic QA systems designed to retrieve information for given queries, such as Ask Jeeves[1]. In addition, more and more user interactive QA systems have been launched in recent years, including Yahoo! Answers[2], Microsoft QnA[3] and BuyAns[4]. These QA systems provide the opportunities for users to post their questions as well as to answer others' questions. With the accumulation of a huge number of questions and answers, some user interactive QA systems may be able to automatically answer users' questions using text-processing techniques. However, due to the complexity of the human languages, most of the current QA systems are difficult to effectively analyze users' free text questions. Hence the accuracy of the question searching, classification and recommendation in these systems is not very satisfactory and the performance of these systems cannot outperform those well-known search engines, such as Google.

To solve these problems, many researchers are engaged in the efforts for improving the capability of machine understanding on questions. (Cowie et al., 2000) use the Mikrokosmos ontology in their method to represent knowledge about the question content as well as the answer. A specialized lexicon of English is then built to connect the words to their ontological meanings. (Hao et al., 2007) propose an approach to using semantic pattern to analyze questions. However, processing of natural language text is complicated especially when a word may have different meanings in different context. For example, given two questions "What are the differences between *Apple* and Dell?" and "What are the differences between *apple* and banana?", the word "*Apple*" in the first question represents a company name while "apple" in the second question refers to a kind of fruit. It is usually difficult for a computer to determine suitable meanings of words under the question context with only several words.

Furthermore, in a real QA system, questions are usually asked in an informal syntax. Some questions are submitted in long sentence while others are posted only with a few words. This kind of irregularity could increases the complexity of

[1] http://uk.ask.com/
[2] http://answers.yahoo.com/
[3] http://qna.live.com/
[4] http://www.buyans.com/

Liu G., Lu Z., Hao T. and Wenyin L.
AUTOMATIC TEXT ANNOTATION FOR QUESTIONS.
DOI: 10.5220/0002796702270236
In *Proceedings of the 6th International Conference on Web Information Systems and Technology (WEBIST 2010)*, page
ISBN: 978-989-674-025-2

analyzing such questions. In a question, keywords are the core semantic units and can be viewed as main point for the given question. If a keyword is misunderstood by the machine, it is hard for the machine to extract right answers from the corpus for this question. Thus, the quality of recognizing and semantically annotating the keywords has significant effect on question understanding and answer retrieval.

Considering the importance of semantics of keywords, in this paper, we propose a new approach to acquiring keywords structures and automatically annotating keywords in questions with semantic labels to facilitate machine understanding. This method first uses a part of speech (POS) tool, such as MiniPar (Lin, 2003), to acquire keywords of a given question. A statistical technique is developed to unambiguously estimate and assign the most appropriate semantic labels for these keywords which contain more than one meaning. We make use of a two-word list named Semantic Labelled Terms (SLT), in which each item records the occurrence of a word's latent semantic labels with the condition that another word occurs at the same time. A naïve Bayesian model is developed to estimate the semantic label of each keyword, with the hypothesis that each word in a sentence is considered to be independently distributed. If there is no corresponding label extracted from SLT, WordNet[5] is then employed to obtain the upper concepts of the keyword by measuring the similarity between the keyword and its candidate labels in a semantic label list defined by the Tagger Ontology mapping table. In addition, an automatic semantic label tagging method is developed to estimate the most semantically related label from the candidates. All keywords in the original question are annotated with semantic labels selected using the above method. In our experiment, we implement our method as a service in our user-interactive QA system – BuyAns. Six groups of words from different domains are chosen to be annotated with semantic labels and their annotated results are also evaluated. Experimental results show that in average 76% annotations are correct according to our evaluation method.

The rest of this paper is organized as follows: we briefly review related work in Section 2. Section 3 introduces the mechanism of the approach proposed in this paper. The experimental results and evaluation are presented in Section 4. Finally, we draw a conclusion and discuss future work in Section 5.

2 RELATED WORK

In the past few years, annotation of documents as a tool for document representation and analysis are widely developed in the field of Information Retrieval (IR). Semantic Annotation is about assigning to the entities in the text links to their semantic descriptions (Kiryakov et al., 2004). Many approaches of semantic annotation are employed for tagging instances of ontology classes and mapping them into the ontology classes in the research of semantic web (Reeve et al., 2005). (Carr et al., 2001) provide an ontological reasoning service which is used to represent a sophisticated conceptual model of document words and their relationships. They use their self-defined data called *metadata* to annotate the web resources. In a webpage, metadata provides links into and from its resources. With metadata, such a web-based, open hypermedia linking service is created by a conceptual model of document terminology. Users could query the metadata to find their wanted resources in the Web. (Handschuh et al., 2002) present the semantic annotation in the S-CREAM project. The approach makes use of machine learning techniques to automatically extract the relations between the entities. All of these entities are annotated in advance. A similar approach is also taken within the MnM (Vargas-Vera et al., 2002), which provides an annotation method for marking up web pages with semantic contents. It integrates a web browser with an ontology editor where semantic annotations can be placed inline and refer to an ontology server, accessible through an API. (Kiryakov et al., 2004) proposed a particular schema for semantic annotation with respect to real-word entities. They introduce an upper-level ontology (of about 250 classes and 100 properties), which starts with some basic philosophical distinctions and then goes down to the most common entity types (people, companies, cities, etc.). Thus it encodes many of the domain-independent commonsense concepts and allows straightforward domain-specific extensions. On the basis of the ontology, their information extraction system can obtain the automatic semantic annotation with references to classes in the ontology and to instances.

In the field of computational linguistics, word sense disambiguation (WSD) in sentence annotation is an open problem, which comprises the process of

[5] http://wordnet.princeton.edu/

identifying which sense of a word is used in any given sentence, in which the word has a number of distinct senses (polysemy). Solution of this problem impacts such other tasks of computation linguistics, such as discourse, improving relevance of search engines, reference resolution, coherence (linguistics), inference and others. These approaches normally work by defining a window of N content words around each word to be disambiguated in the corpus, and statistically analyzing those N surrounding words. Two shallow approaches used to train and then disambiguate are Naïve Bayes classifiers and decision trees. In recent research, kernel based methods such as support vector machines have shown superior performance in supervised learning.

In the application of QA systems, approaches of annotation are developed to analyze text of questions and extract the structure of questions. (Veale, 2002) 0 use the meta-knowledge to annotating a question and generate an information-retrieval query. With this query, the system searches an authoritative text archive to retrieve relevant documents and extracts the semantic entities from these documents as candidate answers to the given question. In his annotation method, non-focal words in a question would be pruned and focus words would be expanded by adding synonyms and other correlated disjuncts. All these possible disjunctions combined by the conjunction operators (e.g. #add, #or) are presented as annotations in stand of the focus word. (Prager et al., 2000) present a technique for QA called Predictive Annotation. Predictive Annotation identifies potential answers to questions in text, annotates them accordingly and indexes them. They extract the interrogative pronouns such as what, where and how long as *Question Type*. They choose an intermediate level of about 20 categories which correspond fairly close to the name–entity types of (Sfihari et al., 1999). Each category is identified by a construct called *QA-Token*. The *QA-Token* serves both as a category label and a text-string used in the process. For example, the query "How tall is the Matterhom" gets translated into the new format of "*LENGTH$*" is the Matterhom. Thus the question is converted into a form suitable for their search engine and then the relative answers are returned to the users. In the question process, all the interrogative pronouns are treated as the *Question Type*. If a question posted is not well-formed or without the interrogative pronoun, their system might fail to process it. Thus it might not flexible for the query analysis process and question representation. (Prager et al., 2001) also propose another method called virtual annotation for answering the *what-is* questions. They extract *Question Type* and target word from a user well-form question. They look up the target word in a thesaurus such as WordNet and use hypernyms returned by WordNet as the answers for the given *what-is* question. To obtain best suitable answer from these hypernyms, they use each hypernym with its target word as the query to search in their database. The hypernym which has the most frequently co-occurring with the target word is selected as the answer. This method is not flawless. One problem is that the hierarchy in WordNet does not always correspond to the way people define the word. Another one causing the error resource is polysemy. In these circumstances, the hypernym is not always suitable for the answer.

In the User-interactive QA field, the mentioned approaches of annotation are not widely used for the text-processing of the questions. Partly because current methods are limited in analyzing informal questions and could not effectively distinguish polysemous keywords in the questions automatically. Therefore, this paper has proposed a new automatic annotation method of identifying and selecting the most related semantic labels for tagging the keywords of the questions. This method employs an effective technique in indicating the word-senses of the polysemous words. Moreover, the new format structure with such semantic annotation is well formed to represent the original question and could be easily recognized and understood by the machine.

3 THE APPROACH

To annotate a free text question, the process of our proposed approach consists of three main modules: keywords extraction module, semantic label selection module and semantic label tagging module. Given a new free text question, the keywords extraction module firstly pre-processes the question using stemming, Part-of-Speech and Name Entity Recognition to acquire all the key nouns (also referred to as keyword). In the semantic label selection module, our system uses keywords as a query to match the records in Semantic Labelled Terms (SLT) to obtain the suitable semantic labels to annotate the keywords extracted in the keywords extraction module.

SLT is built as a kind of semantic dictionary, which uses a formatted two-word list to record the occurrences of two words co-occurred in the same question with their corresponding semantic labels (Hao et al., 2009). SLT consists of two parts: one-

word list and two-word list. In the one-word list, each item contains one word, its corresponding semantic labels and the occurrences of this word tagged by the semantic labels historically. Each element in the one-word list is formatted as follows:
([*Word$_i$*] HAVING [*Semantic_labels*]): Occurrence

On the other hand, the two-word list considers the semantic label to each word in the context of a question. In the two-word list, each item records the occurrences of semantic labels for every pairs of words in a question. We format each element in the two-word list as follows:
([*Word1*] HAVING [*Semantic_labels1*] WITH [*Word2*] HAVING [*Semantic_labels2*]): Occurrence

Where the *Semantic_labels* can be added and the *Occurrence* can be increased and updated when there are new semantic labels used for the current word.

For the keywords in the given free text question, if there are records matched in SLT, the system retrieves the related semantic labels for them. Since some keywords are polysemous and several related records may be matched, the system employs a naïve Bayesian model to select the most relevant semantic label from those candidate records. If the keywords are not matched in SLT, the semantic label tagging module is called, in which each keyword is queried in WordNet to obtain its upper concepts and then corresponding concepts are retrieved with the Tagger Ontology (cf. 3.3). Since all the concepts in this ontology are mapped to WordNet, the related semantic labels in this ontology can be acquired by calculating the similarity between the keyword and each matched concept and finally are used for annotating the keywords of the question. The related workflow is shown in Figure 1.

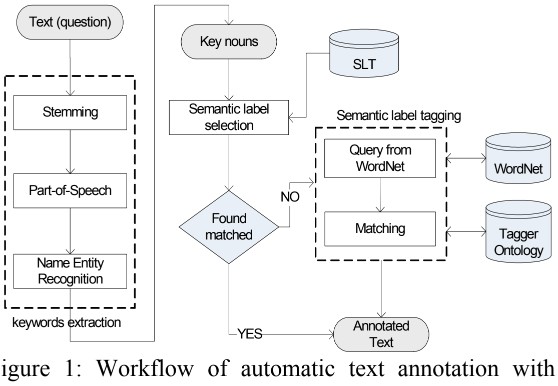


Figure 1: Workflow of automatic text annotation with semantic labels.

3.1 Finding Key Nouns Extraction

Given a new free text sentence, it is important to analyze key nouns, which is the nouns in the main structure of the sentence, by using nature language processing techniques. There are many Part-of-Speech methods and tools such as TreeTagger[6]. Most of these tools identify all the words without considering the importance of them in the sentence. Therefore, the nouns even in attributive clauses are also identified. Such nouns actually decrease the accuracy of the semantic representation of main point in the sentence. In our research, we only consider the nouns in the main structure of a sentence and call them key nouns.

Dependency Grammar (DG) is a class of syntactic theories developed by Lucien Tesnière. The sentence structure is determined by the relation between a word (a head) and its dependents, which is distinct from phrase structure grammars[7]. The dependency relationship in this model is an asymmetric relationship between a word called head (governor) and another one called modifier (Hays, 1964). This kind of relationship can be used to analyze the dependency thus to acquire the main structure and key nouns effectively. MiniPar is a broad-coverage parser for the English language (Lin, 2003). An evaluation with the SUSANNE corpus shows that MiniPar achieves about 88% precision and 80% recall with respect to dependency relationships[8].

Therefore, we use MiniPar to discover and acquire the key nouns by analyzing the dependency relationship. An output of MiniPar mainly consists of three components in the form of “[word, lexicon category, head]”. Figure 2 shows the output with an example of “What is the density of water?”

```
E0       (()      fin C   *         )
 1 (What  ~ N   E0      whn    (gov fin))
 2 (is      be VBE       E0     i        (gov fin))
 E2       (()      what N       4        subj   (gov            density)
   (antecedent 1))
 3 (the     ~ Det 4        det     (gov density))
 4 (density        ~ N   2        pred   (gov be))
 5 (of       ~ Prep         4        mod    (gov density))
 6 (water  ~ N   5        pcomp-n     (gov of))
 7 (?        ~ U   *        punc)
```

Figure 2: Dependency relationship of “*What is the density of water?*” processed by MiniPar.

[6] http://www.ims.unistuttgart.de/projekte/corplex/TreeTagger/

[7] http://en.wikipedia.org/wiki/Dependency_grammar

[8] http://www.cs.ualberta.ca/~lindek/downloads.htm

In this example, the key noun "*density*", which indicates that the asker concerns one property "*density*" of the liquid "*water*", can be acquired firstly in this short text by the dependency grammar. As a result, the word "*density*" is regarded as a key noun for the following process.

3.2 Semantic Label Selection based on Naïve Bayesian Model

Since the high diversity of language expression, a text sentence could be described in many ways and the same word in different contexts would have totally different meanings. Thus annotation of the multiple meaning words is a challenging research work. For better annotating keywords in a text paragraph (e.g. a question) from multiple meanings, we employ a naïve Bayesian formulation with the hypothesis that each word in a question is thought to be independently distributed when determining the semantic label of each word. Given a new question, the system first removes stop words and then acquires all keywords <$Word_{1,}$ $Word_{2,}$... $Word_n$>. For any two words $Word_i$ and $Word_j$, the probability of $Word_i$ assigned with the semantic label $label'$ can be calculated by Equation (1).

$$P(Word_i \rightarrow label' | Word_j) = \frac{P(Word_i \rightarrow label')P(Word_j | Word_i \rightarrow label')}{\sum_{k=1}^{m} P(Word_i \rightarrow label_k)P(Word_j | Word_i \rightarrow label_k)} \quad (i \neq j) \qquad (1)$$

Where $P(Word_i \rightarrow label' | Word_j)$ denotes the probability of Wordi assigned with semantic label $label'$ in the condition that $Word_i$ co-occurs with $Word_j$; $P(Word_i \rightarrow label')$ is the probability of $Word_i$ assigned with semantic label $label'$; $P(Word_j | Word_i \rightarrow label')$ represents the probability of occurring $Word_j$ when $Word_i$ is assigned with $label'$. $\sum_{k=1}^{m} P(Word_i \rightarrow label_k)P(Word_j | Word_i \rightarrow label_k)$ is the prior probability and it is a constant value Hence we only need to calculate the product of $P(Word_i \rightarrow label')$ and $P(Word_j | Word_i \rightarrow label')$ to determine the semantic label of $Word_i$ using the following equation:

$$label^* = \underset{label \in LABEL}{\arg\max} \{P(Word_i \rightarrow label) \times P(Word_j | Word_i \rightarrow label) \qquad (2)$$

For a given $word_i$, $label'$ represents any label in the label set *LABEL*, which refers to all labels in Tagger Ontology. $label^*$ is the most suitable label for the $word_i$. Hence, $word_i$ is annotated by $label^*$ on the condition that $word_i$ co-occurs with $word_j$.

3.3 Tagger Ontology

The fundamental task of the question annotation is to annotate keywords with appropriate semantic labels in a given question. WordNet are large lexical resources freely-available and widely used for annotation (Álvez et al., 2008). It provides a large database of English lexical items available online and establishes the connections between four types of Parts of Speech (POS) - noun, verb, adjective, and adverb. The basic unit in WordNet is synset, which is defined as a set of one or more synonyms. Commonly, a word may have several meanings. The specific meaning of one word under one type of POS is called a sense. Each sense of a word is in a different synset which has a gloss defining the concept it represents. Synsets are designed to connect the word and its corresponding sense through the explicit semantic relations including hypernym, hyponym for nouns, and hypernym and troponym for verbs. Holonymy relations constitute is-a-kind-of hierarchies and meronymy relations constitute is-a-part-of hierarchies respectively.

However, WordNet has too many upper concepts and complicated hierarchy levels for a given concept. Therefore it is difficult to organize and maintain semantic labels in controllable quantity, especially when these semantic labels are used for common users in a user interactive QA system. The concise representations of semantic labels have many advantages such as effectively simplifying the hierarchical structure of ontology as well as reducing complexity of the calculation of similarity between words and labels. Consequently, we propose a Tagger Ontology with only two levels to maintain these semantic labels.

Since the construction of the concept nodes in the ontology is for all open domains, we use a well defined standard taxonom[9] to build the core structure. The ontology is organized as containing certain concepts at the upper levels of the hierarchy of WordNet and it can be mapped to WordNet by a mapping table (samples are shown in Table 1). For better understanding and easy usage by users, it just includes two-level concepts, which have IS_A

[9] http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/definition.html

relationship used to represent hyponymy relationship between two semantic labels.

The semantic labels in the Tagger Ontology are defined as [*Concept 1*] \ [*Concept 2*], where these two concepts *Concept 1* and *Concept 2* have the relationship of *SubCategory*(*Concept1, Concept 2*). Our Tagger Ontology consists of 7 first level concepts and 63 second level concepts in total. Table 1 shows some examples of semantic labels and their corresponding labels in WordNet.

The ontology is mainly used to extract a semantic label of a word in the following way. For a given question, we first obtain its syntactic structure and find all nouns using POS tagger. We then retrieve its super concepts of each noun in WordNet. We finally retrieve these super concepts in the Tagger Ontology to find a suitable semantic label for annotating each of nouns.

For example, for a given free text question "*What is the color of rose?*", the system first analyzes the question and obtains all the nouns "*color*" and "*rose*" by simple syntax-analysis using POS tagger. The super concepts of each noun can be retrieved from WordNet. In this example, the super concepts of "*rose*" are "*bush, woody plant, vascular plant, plant, organism, living thing, object, physical entity, entity*". Among these concepts in WordNet, by mapping with the Tagger Ontology using the mapping table, only "*plant, physical entity*" are acquired. Hence, the semantic label of "*rose*" is tagged as "*[Physical_Entity\Plant]*" finally.

3.4 Semantic Label Tagging based on Similarity

In our user interactive QA system – BuyAns, a mapping table, which represents the bijection between the two-level concepts in our Tagger Ontology and the upper level of hierarchy in WordNet (Miller, 1995), is manually constructed. In Table 1, a partial mapping table is given as an example.

Table 1: Examples of semantic labels and mapped words in WordNet.

Semantic labels	Mapped words in WordNet
[human]\[title]	[abstraction]\[title]
[location]\[city]	[physical_entity]\[city]
[location]\[country]	[physical_entity]\[country]
[location]\[state]	[abstraction]\[state]
[numeric]\[count]	[abstraction]\[count]
[numeric]\[date]	[abstraction]\[date]
[numeric]\[distance]	[abstraction]\[distance]

To assign the best semantic label, we use similarity between words in WordNet and semantic labels in our Tagger Ontology to evaluate the candidate labels. To calculate the similarity, we first employ a traditional distance based similarity measurement (Li et al., 2003), which is shown in equation (3).

$$S(word_i, word_j) = \frac{1}{-\log(\frac{1}{Dis\tan ce}) + 1} \qquad (3)$$

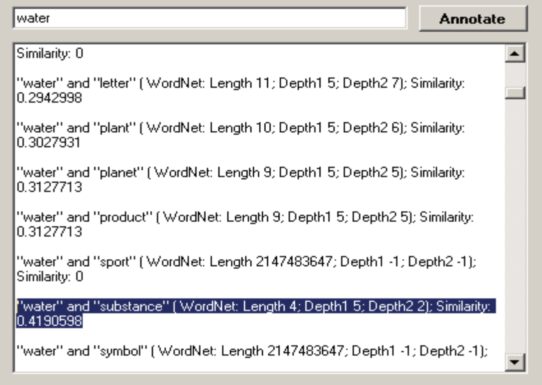
Based on this distance based similarity method, we propose a new similarity measurement considering the word depth in the WordNet hierarchy structure. In this measurement, the semantic labels are mapped to the concepts in WordNet firstly. The similarities between each candidate noun acquired from the question by Minipar and all the concepts already mapped are calculated to find the maximum value. The equation of this measurement is shown as follows.

$$S'(word_i, word_j) = \frac{(Depth_{word_i} + Depth_{word_j})}{36} \times \frac{1}{-\log(\frac{1}{Dis\tan ce}) + 1} \qquad (4)$$

where *Depth* refers to the quantity of concept nodes from the current concept to the top of the lexical hierarchy. *Distance* is defined as the quantity of concept nodes in the shortest path from $word_i$ to $word_j$ in the WordNet. Since the maximum value of *Depth* for the whole hierarchy in WordNet is 18, we use 36 to represent the double value of maximum *Depth*.

Since a semantic label is defined as two related concepts (referred to as Section 3.3), the similarity between a given word and a semantic label can be obtained by representing the semantic label with the concepts. The label with the highest similarity value is selected as the most appropriate label for this word. Figure 3 shows an example of the similarity calculation for the word "*water*".

In this example, the label *"substance"* has the highest similarity in this measurement. Accordingly, the semantic label *"entity\substance"* in our Tagger Ontology is matched with its counterpart *"physical_entity\substance"* in WordNet. Therefore, the semantic labels *"entity\substance"* is assigned to the word *"water"* as the best annotation finally.



```
water

Similarity: 0

"water" and "letter" ( WordNet: Length 11; Depth1 5; Depth2 7); Similarity:
0.2942998

"water" and "plant" ( WordNet: Length 10; Depth1 5; Depth2 6); Similarity:
0.3027931

"water" and "planet" ( WordNet: Length 9; Depth1 5; Depth2 5); Similarity:
0.3127713

"water" and "product" ( WordNet: Length 9; Depth1 5; Depth2 5); Similarity:
0.3127713

"water" and "sport" ( WordNet: Length 2147483647; Depth1 -1; Depth2 -1);
Similarity: 0

"water" and "substance" ( WordNet: Length 4; Depth1 5; Depth2 2); Similarity:
0.4190598

"water" and "symbol" ( WordNet: Length 2147483647; Depth1 -1; Depth2 -1);
```

Figure 3: Similarity calculation for the word “water”.

3.5 Application of Question Annotation

As we have discussed, question annotation can be used for many aspects in QA system, such as question classification and question recommendation. In our system, the annotated questions are mainly used on question classification and pattern based automatic QA.

For the question classification, given a new question *q*, after acquiring *m* semantic labels of key nouns, which are the meaningful nouns obtained by sentence processing, we can calculate the score of each category C_j for each semantic label *Score*(C_j,$Label_n$) by using LCMT (Hao et al., 2009). The number of occurrence of category C_j containing $Label_n$ is also considered in the whole SLT. *Score*(C_j,*q*), the score of each category C_j, for all *m* semantic labels in question *q* is calculated and the scores for all C_j are then compared and the categories are ordered according to their scores to obtain the top *x* categories.

For the pattern based automatic QA, we annotate questions with patterns and semantic labels. For a new question *q*, we can acquire a best matched pattern with Pattern matching technique. After that, since each question is assigned a unique pattern ID in our pattern database, we can acquire related questions and answers easily by query pattern ID in the QA Database with Pattern. For each question in such related question set *QC* (qc_1, qc_2 ... qc_n) we can obtain its key nouns *KNC* (knc_1, knc_2, knc_m) (0<*m*) easily since it is associated with a certain pattern. The similarity $Sim(kn_i,knc_i)$ between each key noun *kn* in *q* and knc_i in *QC* can be calculated. Thus the final similarity between them can be used to identify the most similar questions.

4 EXPERIMENTS AND EVALUATION

To evaluate the proposed method, we develop a Windows application where a question can be annotated with semantic labels automatically. In our system, given a new question, MiniPar is used to identify key nouns. Afterward, with the Tagger Ontology, each noun selected is tagged with a semantic label. Two similarity measurements mentioned above are employed to acquire most appropriate semantic label for each of key nouns. The first similarity measurement only concerns the distance parameter of concepts in WordNet. The second measurement improves the first one by considering depth of concepts. It also takes into account the whole depth of the WordNet hierarchical structure to normalize the similarity value. A user interface of the program including keywords extraction, two similarity measurements, and semantic label tagging is implemented.

Since MiniPar is used to extract keywords for a given question and the evaluation result is already provided in official website, it is unnecessary to test the performance of keywords extraction. In our experiment, we selected different categories of keywords and predefined them with semantic labels manually to build the ground truth dataset for semantic label annotation evaluation.

To evaluate the performance of annotation, the standard measurements such as recall, precision and F1 measures are used. Recall and precision measures reflect the different aspects of annotation performance. Usually, recall and precision have a trade-off relationship: increased precision results in decreased recall, and vice versa. In our experiment, recall is defined as the ratio of correct annotation made by the system to the total number of relevant keywords, which is greater than 0. Precision is defined as the ratio of correct annotations made by the system to the total number of keywords.

$$RECALL = \frac{|Corrected\ \ annotation\ s|}{|R\,elevant\ \ keywords\ |},\qquad PRECISION = \frac{|Corrected\ \ annotation\ s|}{|Total\ \ keywords\ |} \tag{5}$$

In the experiment, since there is no open test data of question annotation available, we choose 6 categories and 50 nouns in each category from the Web as the test data to test the keywords annotation. Most of data are all from Wikipedia[10] and others are

[10] http://en.wikipedia.org/wiki/Word_sense_disambiguation

from open category list (e.g. animal category). Our system automatically annotates these words with semantic labels through two measurements. Since the ground truth in each category has already been defined, the correct annotations can be obtained by comparison of annotated labels and predefined annotations. The experimental results of keywords annotation for these categories with different similarity measurements are shown in Table 2. The average precision and recall for measurement 1 (M1, referred to equation 3) are 72% and 82%, respectively. For measurement 2 (M2, referred to equation 4), the precision and recall are 76% and 86%, respectively. For the category 4 (entity\planet), the annotation result is not very good. It is partly because many planets are named by religious gods like "Tethys" and "Jupiter" such that many of them are annotated as "entity\religion". While in Category 2 (entity\vehicle), there is no description for some words like "quadricycle" and "Velomobile". Thus no annotation is for them. Other words like "toyota" and "benz" are car brands and also cannot find appropriate descriptions in WordNet. In Category 6 (entity\sport), some words like "canoe" and "yacht" are annotated as "entity\vehicle" while "throwing" and "fencing" are annotated as "entity\action".

To better measure the annotation performance, we also use the F1 measure which combines precision and recall measures, treated with equal importance, into a single parameter for optimization. Its definition is presented in equation (6) and its experimental results are shown in Figure 4. From the results, we can see that both two measurements achieve good performance over four categories (C1, C2, C3 and C5). Our proposed measurement 2 has a better performance than that of measurement 1 (traditional distance based method) in annotating the words from all of these categories.

$$F_1 = \frac{2 \times PRECISION \times RECALL}{PRECISION + RECALL} \qquad (6)$$

Given a question set $Q = \{q_1, q_2, \ldots q_m\}$, for each q_i $(1 \le i \le m)$, suppose there are n key nouns in q_i. $S(KN_j)$ $(1 \le j \le n)$ represents whether a key noun KN_j is selected for keyword annotation correctly. $A(KN_j)(1 \le j \le n)$ means whether a key noun is annotated with appropriate semantic label correctly. The values of $S(KN_j)$ and $A(KN_j)$ are either 0 or 1. Therefore, the average annotation precision of q_i can be calculated by equation (7).

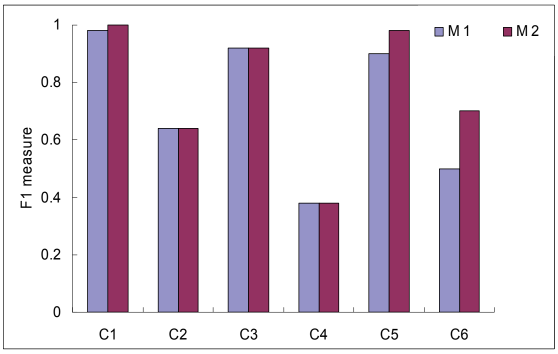


Figure 4: Comparison of annotation performance using F1 measure.

$$PRECISION(q_i) = \frac{1}{n}\sum_{j=1}^{n} S(KN_j) \times A(KN_j) \qquad (7)$$

Since all the key nouns are extracted by MiniPar and the average precision of MiniPar is 88%, which is provided in the official website, we can regard the precision of key nouns selection for annotation as 88%. Therefore, we can calculate the average precision of question annotation and the results are 63.4% and 66.9% using measurement 1 and measurement 2, respectively.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel method to automatically annotate questions with semantic labels. Given a new free text question, the keywords extraction module first processes the question to acquire all the keywords. In the semantic label selection module, we use each keyword as a query to match and retrieve the appropriate semantic labels from the semantic labelled terms (SLT) using a naïve Bayesian method. In the semantic label tagging module, each keyword is assigned with the best label by calculating the similarity between the keyword and each mapped concept in WordNet and the Tagger Ontology. We implement the proposed method and evaluate it with a ground truth dataset. Six categories of nouns are tagged automatically and preliminary results show that the proposed automatic annotation method can achieve a precision of 76% in keywords annotation and 66.9% in question annotation.

Table 2: Experimental results of keywords annotation with two similarity measurements.

		C1	C2	C3	C4	C5	C6	Average
		entity\animal	entity\vehicle	location\country	entity\planet	entity\food	entity\sport	
Precision	M 1	0.98	0.64	0.92	0.38	0.9	0.5	0.72
	M 2	1	0.64	0.92	0.38	0.9	0.7	0.76
Recall	M 1	0.98	0.97	0.94	0.59	0.9	0.54	0.82
	M 2	1	0.97	0.94	0.59	0.9	0.76	0.86

However, some categories such as “planet” are difficult to be annotated precisely as analyzed in the experiments part. There are also some categories need to be improved in recognition of words with multiple senses. In future work, we will intend to investigate and evaluate more accurate and compatible method to identify the meaning of keywords in the given question thus to further improve the overall performance of the proposed method. We will also explore the applications of the proposed method to more tasks, such as question categorization and recommendation.

ACKNOWLEDGEMENTS

We thank Mr. Xiaojun Quan for his comments and suggestions on this work.

REFERENCES

Cheng, P.J., Chiao, H.C., Pan, Y.C. and Chien, L.F. 2005. Annotating text segments in documents for search. *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 317- 320.

Hao, T.Y., Hu, D.W., Liu, W.Y. and Zeng, Q.T. 2007. Semantic patterns for user-interactive question answering, Journal of Concurrency *and Computation: Practice and Experience* 20(1), 2007.

Lin, D. 2003. Dependency-based evaluation of MINIPAR. *Treebanks: Building and Using Parsed Corpora*, 2003.

Prager, J., Brown, E. and Coden, A. 2000. Question-answering by predictive annotation, *Proceedings of the 23rd Annual International ACM SIGIR conference,* Athens, 2000.

Sfihari, R. and Li, W. 1999. Question answering supported by information extraction, *Proceedings of the Eighth Text REtrieval Conference (TREC8)*, Gaithersburg, Md., 1999.

Carr, L., Bechhofer, S., Goble, C. and Hall, W. 2001. Conceptual linking: ontology-based open hypermedia, *Proceedings of the 10th International World Wide Web Conference*, pp. 334–342, Hong Kong, 2001.

Handschuh, S., Staab, S. and Ciravegna, F. 2002. S-CREAM - semiautomatic creation of metadata, *Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW 2002)*, Springer Verlag, 2002.

Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A. and Ciravegna, F. 2002. MnM: ontology driven semi-automatic and automatic support for semantic markup, *Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW 2002)*, Springer Verlag, 2002.

Kiryakov, A., Popov,B., Ognyanoff,D., Manov, D. and Goranov, K.M. 2004. Semantic annotation, indexing, and retrieval, *Journal of Web Semantics*, pp. 49–79, 2004.

Reeve, L. and Han H. 2005. Survey of semantic annotation platforms, *Proceedings of the 2005 ACM Symposium on Applied Computing*, Santa Fe, New Mexico, March 13 - 17, 2005.

Veale, T. 2002. Meta-knowledge annotation for efficient natural-language question-answering, *Proceedings of the 13th Irish International Conference (AICS 2002)*, Limerick, Ireland, pp. 115-128, September 12-13, 2002.

Prager, J., Radev D. and Czuba K. 2001. Answering what-is questions by virtual annotation, *Proceedings of the first International Conference on Human Language Technology Research 2001*, San Diego, March 18 - 21, 2001.

Hays, D. 1964. Dependency theory: a formalism and some observations, *Language*, Linguistic Society of America, Vol. 40, No. 4, pp. 511-525, 1964.

Miller, G. A. 1995. WordNet: a lexical database for English, *Communications of the ACM*, Vol. 38, Issue 11, 1995.

Li, Y.H., Bandar, Z.A. and McLean, D. 2003. An approach for measuring semantic similarity between words using multiple information sources, *IEEE Transactions on Knowledge and Data Engineering* Vol. 15, No. 4, July/August, 2003.

Cowie, J., Ludovik, E., Molina-Salgado, H., Nirenburg, S. and Sheremetyeva, S. 2000. Automatic question answering, *Proceedings of the Rubin Institute for Advanced Orthopedics Conference*, Paris, 2000.

Álvez, J., Atserias, J., Carrera, J., Climent, S., Laparra, E., Oliver, A. and Rigau, G. 2008. Complete and consistent annotation of wordNet using the top concept ontology. *Proceedings of Sixth International Language Resources and Evaluation (LREC'08)*, European Language Resources Association (ELRA), 2008.

Hao, T.Y., Ni, X.L., Quan, X.J., W.Y. Liu 2009. Automatic Construction of Semantic Dictionary for Question Categorization, *Proceedings of The 13th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2009*, Orlando, pp. 220-225, July 10-13, 2009.