Automatic Construction of Semantic Dictionary for Question Categorization

Tianyong HAO
Department of Computer Science, City University of Hong Kong
Hong Kong, China

Xingliang NI
Department of Computer Science and Technology, University of Science & Technology of China
Hefei, China

Xiaojun QUAN
Department of Computer Science and Technology, University of Science & Technology of China
Hefei, China

Wenyin LIU
Department of Computer Science, City University of Hong Kong
Hong Kong, China

ABSTRACT

An automatic method for building a semantic dictionary from existing questions in a pattern-based question answering system is proposed for question categorization. This dictionary consists of two main parts: Semantic Domain Terms (SDT), which is a domain specific term list, and Semantic Labeled Terms (SLT), which contain common terms tagged with semantic labels. The semantic dictionary is built using the proposed method on a set of 2509 questions with semantic patterns in our system. 3390 questions without semantic patterns are used as ground truth to test its performance. Experimental results show that the precision of question classification is improved by 7.5% in average after using the constructed semantic dictionary compared with the baseline method.

Keywords: Semantic Dictionary, Question Classification, Semantic Pattern, SDT, SLT.

1. INTRODUCTION

Automatic question answering (QA) targets at providing more concise and precise answers to users' questions than search engines. Most of the QA systems are difficult to effectively analyze a user's free text question due to the complexity of human languages [1]. Question categorization, which classifies a question into one or more pre-defined categories based on its content, plays an important role in both automatic and user-interactive QA systems. Question categorization also can be used to locate the exact answers in an automatic QA system. In addition, it can help automatically organize questions as well as provide a more convenient way for users to post and browse questions in a user-interactive QA system (also referred to as community based QA system). In order to accurately categorize these questions, we usually need to understand the main topics so that the questions with the same topic are organized into the same category.

Techniques are proposed to improve the performance of text categorization. Shehata & Karray [2] and later Wermter & Hung [3] use WordNet to change document representation from a bag of words to a bag of synsets by using the hypernymy relation to generalize word senses. However, in certain categorization tasks, using the only synsets is not enough and some words in the synsets may are useless. Keyword extraction has also been studied to improve categorization such as Hulth [4], which does comparison on different ways of keyword representation.

Although document categorization has been intensively investigated, question categorization is still a rather different issue. It differs from document categorization due to specific properties of short text questions compared with long text documents. The similarity between documents is usually computed based on the co-occurrence of words they share. Such measurement performs poorly in question similarity calculation since users may use different expressions and similar/relevant questions are usually too short to have sufficient common features. Several literatures have been proposed for leveraging accuracy of question categorization with the feature vector method. For example, Song et al. [1] proposed a model for automatic question categorization by extending each question to a feature space in terms of similar words before calculating its similarity to each category. However, their method takes into account only word similarity when calculating a question's similarity to a category. The semantic properties of a word, such as the super concept of this word, are not used to calculate the probability of a word's domain. We believe these semantic properties can help better understand the main topics of the question and more accurately categorize the question.

We propose to use a semantic dictionary, which contains words' semantic labels and categories, to help enrich the semantics of words and help improve the performance of question classification. We also propose an automatic method for building such semantic dictionary from existing questions in our pattern based question answering system [5]. The semantic dictionary mainly consists of two main parts: Semantic Domain Terms (SDT) and Semantic Labeled Terms (SLT). A Tagger Ontology which defines all semantic labels with two levels is also presented. Based on this ontology, all semantic labels are used for training to obtain a Label-Category Mapping Table (LCMT) from pattern based questions.

We implemented the proposed method in our QA system - BuyAns [6] to build such semantic dictionary and use it to categorize questions compared with Song's method [1], as the baseline. 2509 questions with semantic patterns in the system have been used to build semantic dictionary and 3390 questions without semantic patterns are used as ground truth to test its performance. Experimental results show that the performance of our method improves by 7.5% after using this semantic dictionary.

The rest of this paper is organized as follows: Section 2 describes the proposed semantic dictionary in detail. Section 3 presents the automatic method including automatic semantic domain terms building, automatic semantic labeled terms building and semantic-category mapping table building. Section 4 described the application of semantic dictionary for question classification and implementation in detail. In Section 5, experimental results are shown and Section 6 summarizes this paper and discusses future work.

2. SEMANTIC DICTIONARY

Admittedly, question understanding and categorization can be more efficient if more semantic properties can be known. Based on this assumption, we propose to use ontology to organize terms with their semantic labels (as annotations) into a dictionary. Since the semantic labels are well managed and the relationships between these semantic labels are represented clearly in the ontology way, we name this dictionary as semantic dictionary.

Our semantic dictionary mainly consists of four parts: Semantic Domain Terms (SDT) and Semantic Labeled Terms (SLT) are two main parts, while the Tagger Ontology and Label-Category Mapping Table (LCMT) are two supplemental modules for defining relationships between the semantic labels and mapping semantic labels to categories, respectively. The structure of the semantic dictionary is shown in Fig. 1.

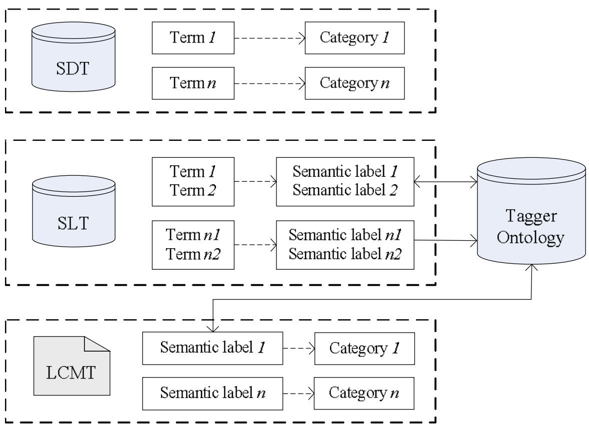

Fig. 1. Structure of the semantic dictionary

SDT, SLT and LCMT

SDT, SLT are two important parts of our semantic dictionary. In SDT, the words are domain-specific terms, which are tagged with their domains. For example, term "*machine gun*" tagged as belonging to the "*military*" category. The format of words and their categories in SDT is as follows:

[*Word$_i$*] HAVING [*Category$_j$*]

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_label$_k$*]): *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 contains the possibility of semantic labels for every pairs of words in a question. We format each element in the two-word list as follows:

([*Word$_1$*] HAVING [*Semantic_label$_1$*] WITH [*Word$_2$*] HAVING [*Semantic_label$_2$*]): *Occurrence*

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

Label-Category Mapping Table (LCMT) is a table mapping semantic labels to their possible content categories. It also records the number of questions which contain such semantic label and belong to a certain category. Each item in the LCMT has the format as follows:

[*Semantic_label$_j$*] **MAP_TO ([*Category$_1$*:*Occurrence$_1$*] ... [*Category$_n$*:*Occurrence$_n$*])**

Where *Occurrence$_n$* denotes the frequency of questions which contain *Semantic_label$_j$* and belong to *Category$_n$*.

Tagger Ontology

These semantic labels are used in not only the semantic dictionary but also other applications such as semantic pattern. Therefore, it is important to define and manage these labels. Tagger Ontology is designed in two levels to manage these labels, in which IS_A relationship is used to represent the relationship between two semantic labels. It is organized as containing certain concepts hierarchy and can be mapped to WordNet [13] by a mapping table (as shown in Table 1).

Table 1. Semantic labels and corresponding labels in WordNet

Semantic labels	Concepts mapped 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*]

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

The ontology is mainly used to extract the appropriate semantic label in the following way. For a given question, we first obtain its syntactic structure and find nouns in it using POS tagger. We then retrieve the super concepts of each noun/verb. We finally search these super concepts in the Tagger Ontology to find suitable semantic labels to tag each of noun/verb.

This Tagger Ontology can be used to provide more semantic information when analyzing questions. For example, when a user posts a question "*What is the color of rose?*" the system first analyzes the question and obtains the question type by finding the type word "*What*". The nouns "*color*" and "*rose*" are also obtained by simple syntax-analysis using POS tagger. Their super concepts can also be found from WordNet. For this example, the super concept of "*rose*" is "*bush, woody plant, vascular plant, plant, organism, living thing, object, physical entity, entity*". Among these concepts, only "*plant, physical entity*" are found in the Tagger Ontology. Hence, the semantic label of "*rose*" is tagged as "[*Physical_Entity\Plant*]" finally.

3. AUTOMATIC CONSTRUCTION OF THE SEMANTIC DICTIONARY

We propose a method for automatically building the semantic dictionary. It consists of two main modules: (1) building Semantic Domain Terms (SDT), (2) building Semantic Labeled Terms (SLT). In the first module, system mainly retrieves the representative terms in different field from the Web, such as Wikipedia. In the second module, system mainly retrieves keywords in our QA system and its corresponding semantic labels in the Tagger Ontology, which are then mapped to the categories in our QA system [5] for training to obtain the Label-Category Mapping Table (LCMT). This table can be used to classify a given semantic label to corresponding categories. The related workflow is shown in Fig. 2.

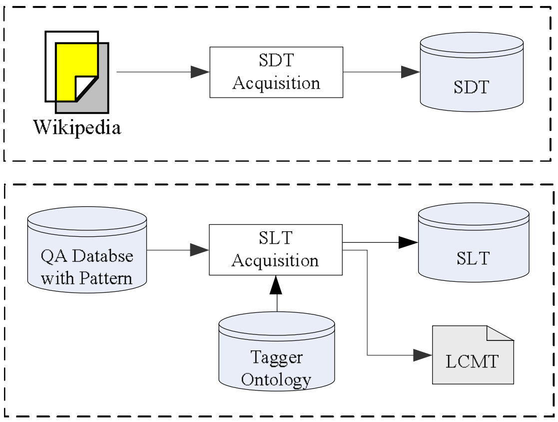

Fig. 2. The workflow of automatic construction of the semantic dictionary

The semantic patterns based questions are important sources for us to construct the semantic dictionary. In our system, a semantic pattern is an extension of the structural pattern which is a generalization of a group of questions which have similar structure. The structural pattern has been demonstrated that it can facilitate machine understanding [12]. We extend the idea of the structural pattern by tagging certain structural elements with semantic labels and name it semantic pattern [15]. When a user submits a new question, the system automatically recommends semantic patterns based on its sentence structure and its content. For example, if a user wants to post a question: "*Who is CEO of IBM?*" our QA system may return a semantic pattern "*<Q>Who<Q> is [Human\Title] of [Entity\Organization]?*" to the user. The user may fill in the placeholders "*[Human\Title]*" and "*[Entity\Organization]*" with "*CEO*" and "*IBM*", respectively. We can know that "*CEO*" is a title of a person and "*IBM*" is an organization from this pattern.

Building Semantic Domain Terms (SDT)

Domain-specific thesauri are high-cost, high-maintenance, high-value knowledge structures. Appropriate usage of these thesauri can leverage the performance of question classification efficiently.

Wikipedia was launched in 2001 with the goal of building free encyclopedias in all languages. Today it outstrips all other encyclopedias in size and coverage, and is one of the most popular sites on the Web. Out of more than three million articles in 125 different languages, one-third is written in English, yielding an encyclopedia almost ten times as big as the Encyclopedia Britannica, its closest rival [16].

We build Semantic Domain Terms (SDT) mainly based on Wikipedia. Firstly, we retrieve the related Web pages for a certain domain name. For instance, we crawls the related pages of domain name "*military*" from Wikipedia [18] with a page crawler tool. After that, we can analyze the HTML pages and parse the tag of source code to obtain some special tags which may represent domain terms. For examples, the tags of "*class = NavFrame*" and "*class = NavHead*" represent the navigation frame and navigation head, in which we can retrieve class names. By these names we can obtain the related domain terms. After collecting these domain terms, we maintain the SDT by computing the correlation between these terms and corresponding domain names by means of search engines. Mutual information is a good measure of independence [17] and can be applied to find the independent level of two words. The mutual information of any two words x and y can be computed by Eq. (1) where mutual information compares the probability of observing x and y together (the joint probability) with the probabilities of observing x and y independently.

$$I(x, y) = \log_2 \frac{p(xy)}{p(x)p(y)} \tag{1}$$

Using this method, we retrieve the related short passages in the top 10 pages by searching domain terms from Google, such as searching "*weapon*". After that, we calculate the probability of "*weapon*" and its domain name "*military*" in all passages. The joint probability of both words is calculated by the same statistical method. Those terms with the mutual information below a threshold are removed out of our dictionary.

Building Semantic Labeled Terms (SLT)

In our QA system, some questions are associated with semantic patterns. Our semantic patterns provide a definite semantic label for each placeholder in them, which distinguishes our semantic patterns from other question templates. Many users post questions with matched semantic patterns everyday and we have accumulated many questions with patterns. Therefore, we can use these questions with semantic patterns to train and learn the semantic labels of these terms.

The procedure of constructing SLT mainly includes the following steps: First we extract terms in all questions and their corresponding semantic labels with the help of the semantic patterns. In this case, if the question is associated with a semantic pattern, we can retrieve the semantic labels of terms according to the position of these terms and the corresponding semantic labels. Secondly, we obtain the co-occurrence of different terms and their corresponding semantic labels in the same question. Finally, we create each entry of the one-word list the SLT with the follow three elements:

$$< Term, Label\_of\_Term, Occurrence >$$

And the two-word list with the following five elements:

$$< Term_{1,} Term_2, Label\_of\_Term_1, Label\_of\_Term_2, Occurrence >$$

The construction of the one-word list is trivial since we only need to count the occurrence of each term with a specific label. As to the two-word list, we take the context of each key noun in a question into consideration. For example, a question "*Which computer sells better apple or dell?*" associated with the pattern "*<Q>Which</Q> computer sells better [entity\product] or [entity\product]?*" "*apple*" and "*dell*" are assigned with the semantic label "*entity\product*". Meanwhile, the question "*Which fruit do you prefer apple or banana?*" is captured by another pattern "*<Q>Which</Q> fruit do you prefer [entity\fruit] or [entity\fruit]?*" In this case, "*apple*" and "*banana*" are assigned with the semantic label "*entity\fruit*". The representation of two-word list in this example is shown in table 2.

Table 2. A example of terms in two-word list

$Term_1$	$Term_2$	$Label\ of\ Term_1$	$Label\ of\ Term_2$	Occurre-nce
Apple	Dell	Entity\product	Entity\product	1
Apple	Banana	Entity\fruit	Entity\fruit	1

Since we have all the questions with semantic labels and corresponding categories, we can train these questions to obtain the Label-Category Mapping Table (LCMT) from semantic labels to categories. The purpose is that: given a new question, the system can assign category according to its semantic labels with the mapping table.

The framework of building mapping table is as follows: for all the questions and their related categories, we acquire all the semantic labels and therefore we can assign the categories for each semantic label. Some labels may have different categories such as "*location\city*" may be assigned to categories of "*China*", "*Tour*" or "*Europe*". We train this automatically and record the number of questions which contain this semantic label and are classified to a certain category before. The format of LCMT is like "[$Category_1$: $Occurrence_1$; … $Category_n$: $Occurrence_n$]", which can benefit to the calculation of probability for each category.

4. APPLYING SEMATNIC DICTIONARY FOR QUESTION CLASSIFICATION

After constructing the semantic dictionary, we can apply it into question classification to increase its performance. Given a new question, we firstly process the question by using stemming method, Part-of-Speech method and Name Entity Recognition method to obtain all keywords. System then queried these keywords from the SDT and SLT. If any keyword is matched, system can obtain related semantic labels and use LCMT to calculate most relevant categories. Otherwise, system will use Song's method [1] to obtain recommended categories. The related workflow is shown in Fig. 3.

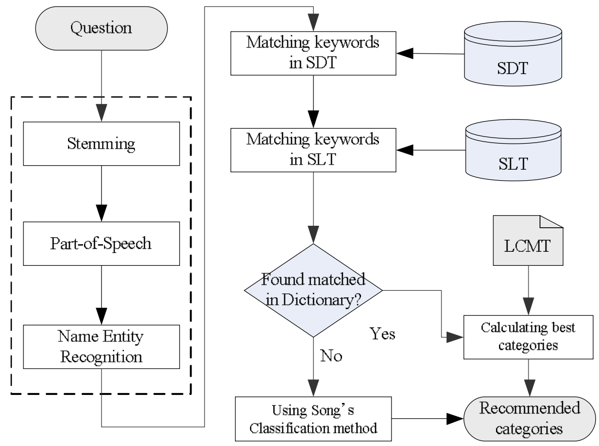

Fig. 3. The workflow of application of dictionary for question classification

With the SLT dictionary, the semantic labels of terms in a given new question can be determined. We employ a naïve Bayesian formulation with the hypothesis that each term in a sentence is thought to be independently distributed to determine the semantic label of each term. If the question is not posted by using semantic pattern, system first remove stop words in this question and then represent it by a vector < $Term_1$, $Term_2$, …, $Term_n$>. The probability of semantic label for each term can be calculated by Eq. (2).

$$P(Term_i \rightarrow label' | Term_j) = \frac{P(Term_i \rightarrow label')P(Term_j | Term_i \rightarrow label')}{\sum_{k=1}^{m} P(Term_i \rightarrow label_k)P(Term_j | Term_i \rightarrow label_k)} \quad (i \neq j) \tag{2}$$

where $P(Term_i \rightarrow label' | Term_j)$ denotes the probability of $Term_i$ assigned with semantic label $label'$ in the condition that $Term_i$ co-occurs with $Term_j$; $P(Term_i \rightarrow label')$ is the probability of $Term_i$ assigned with semantic label $label'$; $P(Term_j | Term_i \rightarrow label')$ represents the probability of occurring $Term_j$ when $Term_i$ is assigned with $label'$. $\sum_{k=1}^{m} P(Term_i \rightarrow label_k)P(Term_j | Term_i \rightarrow label_k)$ is the prior probability and it is a constant value, we only need to calculate the product of $P(Term_i \rightarrow label')$ and $P(Term_j | Term_i \rightarrow label')$ to determine the semantic label of $Term_i$ using Eq. (3).

$$label^* = \mathop{Arg\ max}_{label' \in LABEL} \{P(Term_i \rightarrow label') \times P(Term_j | Term_i \rightarrow label')\} \quad (3)$$

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

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, which is shown in Eq. (4),

$$Score(C_j, Label_n) = \frac{Occurrence\ (C_j)}{\sum_{i=1}^{l} Occurrence\ (C_i)} \quad (4)$$

Where $Occurrence(C_j)$ is the number of occurrences of category C_j containing $Label_n$. $\sum_{i=1}^{l} \mathrm{Occurrence}(C_i)$ is the total number of occurrences for all the *l* categories. The score of each category C_j for all *m* semantic labels in question *q* $Score\ (C_j, q)$ is the sum of all labels' scores, as shown in Eq. (5). The scores $Score(C_j, Label_n)$ for all C_j are then compared and the categories are ordered according their scores to obtain the best categories.

$$Score(C_j, q) = \sum_{k=1}^{m} Score(C_j, Label_n) \quad (5)$$

We implement the proposed method using C# and build the semantic dictionary automatically. The program loads all the questions with semantic patterns and processes them with the Tagger Ontology to build SLT automatically. After that, we can generate the Label-Category Mapping Table (LCMT) according to the semantic labels in each question and its corresponding category as mentioned in previous section. The program also employs a Name Entity (NE) dictionary with which we can tag the name entities such as locations and human names in the questions. A user interface of the program including building SDT, SLT and LCMT is implemented.

With these dictionaries, we design the module to categorize a coming question. After word segmentation, we can first acquire all nouns from the question and add them into a keyword list. These nouns are then tagged by SDT and SLT to fetch the category tags and semantic label tags of them respectively. Based on these tags, the question is assigned to a few candidates of categories with scores.

5. EXPERIMENTS AND EVLAUATION

We obtain 1059 terms from Wikipedia directly to construct the SDT and select 2509 questions with semantic patterns from our pattern-based user-interactive QA system - Buyans [5][6] to construct the SLT. The SLT finally consists of 1382 one-word items and 603 two-word items. To evaluate the performance of our question classification method, which introduces semantic dictionary into classification, we select 3390 questions without semantic patterns from BuyAns with pre-defined categories as ground truth. We first implement Song's method [1] as baseline and then combine the result returned by our semantic dictionary with his method using Eq. (6).

$$Score_i = \lambda \times Score\_SD_i + (1-\lambda) \times Score\_Song_i \quad (6)$$

Where $Score_i$ denotes the final score of category i, $Score\_SD_i$ denotes the score of category i returned by semantic dictionary method and $Score\_Song_i$ denotes the score of category i returned by Song's method respectively. In addition, l is a trade-off factor adjusting the balance between the result of Song's method and ours method.

Table 3. Comparison of the accuracy of our method with baseline

P@n	P@1	P@2	P@3
Song's Method	56.2%	69.3%	74.2%
Our Method	63.5%	76.7%	82.1%

Table 3 shows the experimental results when l is set to 0.5, where P@n is the performance metric in our experiments and also used by Song et al.[1]. It is a popular metric in the IR area. In our paper, it means the proportion of questions whose correct category is within the top n categories our system suggests. From the experimental result, we can see that the accuracy increases 7.3%, 7.4% and 7.9% under P@1, P@2 and P@3, respectively.

In second experiment, we use all 6631 questions from BuyAns to test the performance of semantic dictionary. We assume classifying question correctly is the question with any correct category label. For SDT, totally 934 questions are classified and 673 questions among them are classified correctly. For SLT, 6629 questions are classified and 5586 are classified correctly. For combination of SDT and SLT, 6629 questions are classified and 6114 questions are classified correctly. The accuracy for these three methods is 72.1%, 88.8% and 92.2%, respectively, which are shown in Table 4.

Table 4. The accuracy by using different parts of the semantic dictionary

	# Questions classified	# Questions classified correctly	Accuracy
SDT	934	673	72.1%
SLT	6629	5586	88.8%
SDT+SLT	6629	6114	92.2%

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new automatic method to build semantic dictionary from pattern based question answering system to improve the performance of question classification. The obtained semantic dictionary mainly consists of Semantic Domain Terms (SDT) and Semantic Labeled Terms (SLT). A Tagger Ontology which contains all semantic labels with two categories is also presented. Based on this ontology, all semantic labels are trained to obtain a mapping table from Label-Category Mapping Table (LCMT).

We implemented the proposed method in our QA system. Based on Song's method, we added our dictionary to enhance the semantics of categorization. Experimental results with same test set shows that the performance of categorization using this dictionary is improved 7.5% in average.

This method can be easily applied in user-interactive QA system as well as other question classification system since the dictionary is independent. However, training the questions and constructing the semantic dictionary is a difficult work since it needs a lot of questions with semantic patterns. In the future, we will try the automatic annotation method to tag questions with semantic labels automatically.

7. REFERENCES

[1] Song, W.P., Liu W.Y., Gu, N.J., and Hao, T.Y. "Automatic Categorization of Questions in a User-interactive Question Answering System", The first WWW Workshop on Question Answering on the Web (QAWeb 2008), Beijing, China, April 2008.

[2] Shehata, S., Karray, F. and Kamel, M. "Enhancing Text Retrieval Performance using Conceptual Ontological Graph". In ICDM Workshops, pp. 39-44, 2006.

[3] Wermter, S. and Hung, C. "Selforganizing Classification on the Reuters News Corpus". In Proceedings of COLING, 2002.

[4] Hulth, A. and Megyesi, B.B. A Study on Automatically Extracted Keywords in Text Categorization. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL, 2006.

[5] Hao, T.Y., Song, W.P., and Liu, W.Y. "Automatic generation of semantic patterns for user-interactive question answering", In Proceedings of Asia Information Retrieval Symposium 2008, Harbin, January 16-18, 2008.

[6] BuyAns, http://www.buyans.com. 2007.

[7] Wu, H.R., Phang, T.H., Liu, B. and Li, X. "A Refinement Approach to Handling Model Misfit in Text Categorization". In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.

[8] Shehata, S., Karray, Fakhri. and Kamel, M. "A Concept-based Model for Enhancing Text Categorization". In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007.

[9] Shehata, S., Karray, F. and Kamel, M. "Enhancing Text Clustering using Concept-based Mining Model". In Proceedings of ICDM, pp. 1043-1048, 2006.

[10] Suzuki, J., and Taira, H. Yutaka Sasaki, and et al. "Question Classification using HDAG Kernel". In Proceeding of 6th Information-Based Induction Sciences, pp. 217-222, 2003.

[11] Zhang, D., and Lee, W.L. "Question Classification using Support Vector Machines". In Proceedings of the 26th annual international ACM SIGIR conference of Research and development in information retrieval, pp. 26-32, 2003.

[12] Deepak, R., and Eduard, H. "Learning Surface Text Patterns for a Question Answering System". In Proceedings of the 40th ACL Conference. Philadelphia, 2002.

[13] WordNet, http://wordnet.princeton.edu/. 2007.

[14] Cowie, J., Ludovik, E., Molina-Salgado, H., Nirenburg, S., and Sheremetyeva, S. "Automatic Question Answering". In Proceedings of the Rubin Institute for Advanced Orthopedics Conference. Paris, 2000.

[15] Hao, T.Y., Hu, D.W., Liu, W.Y., and Zeng, Q.T. "Semantic Patterns for User-Interactive Question Answering". Journal of Concurrency and Computation: Practice and Experience, Vol. 20(7), pp.783-799, May 2008.

[16] Milne, D., Medelyan, O., and Witten, I.H. "Mining domain-specific thesauri from wikipedia: A case study". In Proceedings of ACM International Conference on Web Intelligence. pp. 442–448, 2006.

[17] Mutual information and Kullback-Leibler divergence. http://www.cis.hut.fi/aapo/papers/NCS99web/node26.html, 2009

[18] Wikipedia. http://en.wikipedia.org/wiki/Weapon, 2009.