Minimally Supervised Question Classification and Answering based on WordNet and Wikipedia

張至 Joseph Chang 顏孜羲 Tzu-Hsi Yen

蔡宗翰 Richard Tzong-Han Tsai*

元智大學資訊工程學系

Department of Computer Science and Engineering,

Yuan Ze University, Taiwan

{s951533, s940635}@mail.yzu.edu.tw, thtsai@saturn.cse.yzu.edu.tw

*corresponding author

摘要

在此篇論文中，我們提出一個自動將問題分類至現有詞網 (WordNet) 中之細分類方法。為此，我們利用維基百科之特性以及其文章標題，建立大規模語意實體分類表，包含了1,581,865個實體。為了表現我們研究之效用，我們建構了一個基於冗餘原則的自動問題回答系統，並透過所提出的問題分類方法來增進其效能。實驗結果顯示所提出的方法能夠有效地提升問題分類與回答的精確率。

Abstract

In this paper, we introduce an automatic method for classifying a given question using broad semantic categories in an existing lexical database (i.e., WordNet) as the class tagset. For this, we also constructed a large scale entity supersense database that contains over 1.5 million entities to the 25 WordNet lexicographer's files (supersenses) from titles of Wikipedia entry. To show the usefulness of our work, we implement a simple redundancy-based system that takes the advantage of the large scale semantic database to perform question classification and named entity classification for open domain question answering. Experimental results show that the proposed method outperform the baseline of not using question classification.

關鍵詞： 自動問題回答，問題分類，辭彙語意資料庫，辭網，維基百科

Keywords: question answering, question classification, semantic category, WordNet, Wikipedia.

1. Introduction

Question classification is considered crucial to the question answering task due to its ability

to eliminating answer candidates irrelevant to the question. For example, answers to person-questions (e.g., *Who wrote Hamlet?*) should always be a person (e.g., *William Shakespeare*). Common classification strategies includes semantic categorization and surface patterns identification. In order to fully benefit from question classification techniques, answer candidates should be classified the same way as questions.

Surface patterns identification methods classifies questions to sets of word-based patterns. Answers are then extracted from retrieved documents using these patterns. Without the help of external knowledge, surface pattern methods suffer from limited ability to exclude answers that are in irrelevant semantic classes, especially when using smaller or heterogeneous corpora.

An other common approach uses external knowledge to classify questions to semantic types. In some previous QA systems that deploy question classification, named entity recognition (NER) techniques are used for selecting answers from classified candidates. State-of-the-art NER systems produce near human performances. Good results are often achieved by handcrafted complex grammar models or large amount of hand annotated training data.

However, most high performance NER systems deal with a specific domain, focus on homogeneous corpora, and support a small set of NE types. For example, in the Message Understanding Conference 7 (MUC-7) NER task, the domain is "Airplane crashes, and Rocket/Missile Launches" using news reports as the corpus. There are only three NE classes containing seven sub classes: ORG, PERSON, LOCATION, DATE, TIME, MONEY, PERCENT. Notice that in the seven subclasses, only three of them are NEs of physical objects, others are number based entities. This is apparently insufficient for candidates filtering for general question answering. Owing to the need of wider range NE types, some of the later proposed NE classes construct of up to 200 sub classes, but NER systems targeting these types of fine-grained NE classes may not be precise enough to achieve high performance.

The amount of supported classification types greatly influences the performance of QA systems. A coarse-grained classification achieving higher precision, may still be weak in excluding improper answers from further consideration. A fine-grained classification may seem a good approach, but the cost of high-precision classification may be too high to produce actual gain in QA systems.

Moreover, in open domain QA, answers are not necessarily NEs nor can they be captured by using simple surface patterns. Using a small set of NE types to classify questions has its limits. We randomly analyzed 100 question/answer pairs from the Quiz-zone Web site (http://www.quiz-zone.co.uk/), only 70% of them are NEs. This shows being able to classify common nouns is still very important in developing QA systems.

In order to support more general question anwering, where the answer can be NEs and common nouns, we took the approach of using finer-grained semantic categories in an existing lexical database (i.e., WordNet). WordNet is a large scale, hand-crafted lexical ontology database widely used in solving natural language processing related tasks. It provides taxonomy of word senses and relations of 155,327 basic vocabularies that can be used as an semantic taxonomy for entity classification. However, in the later sections of this paper, we will show that WordNet leave room for improvement in question classification and

answer validation, and more entities, especially NEs, are needed to achieve reasonable coverage for answer candidates filtering.

With this in mind, we turn to Wikipedia, an online encyclopedia compiled by millions of volunteers all around the world, consisting articles of all kinds. It has become one of the largest reference tool ever. It is only natural that many researchers have used Wikipedia to help perform the QA task.

Using WordNet semantic categories and rich information from Wikipedia, we propose an minimally supervised question classification method targeting at the 25 WordNet lexicographer's files for question classification. Experimental results show promising precision and recall rates. The method involve extending WordNet coverage and producing the training data automatically from question/answer pairs, and training a maximum entropy model to perform for classification.

The rest of the paper is organized as follows. In the next section, we review related work in question classification and question answering. In Section 3 we explain in detail the proposed method. Then, in Section 4 we report experimental results and conclude in Section 5.

2. Related Work

Text Retrieval Conference (TREC) has been one of the major active research conferences in the field of question answering. The early tasks in the question answering track in TREC focuses on finding documents that contain the answer to the input question. No further extraction of exact answers from the retrieved documents is required.

In an effort to foster more advanced research, the TREC 2005 QA Task focuses on systems capable of returning exact answers rather than just the documents containing answers. Three types of questions are given, including FACTOID, LIST, and OTHER. For every set of questions a target text is also given as the context of the set of questions. LIST questions require multiple answers for the topic, while FACTOID questions required only one correct answer. Therefore, many consider LIST questions are easier.

More recent TREC QA Tasks focuses on complex, interactive question answering systems (ciQA). In ciQA Tasks, fixed-format template questions are given (e.g. What evidence is there for transport of [drugs] from [Mexico] to [the U.S.]?). Complex questions are answerable with several sentences or clauses. (e.g. United States arrested 167 people - including 26 Mexican bankers) The design of an interactive query interface is also a part of this task. In this paper, we focus on the issue of classifying questions in order to effectively identify potential answers to FACTOID and LIST questions.

More specifically, we focus on the first part of question answering task, namely identifying the semantic classes of the question (and answer) that can be used to formulate an effective query for document retrieval and to extract answers in the retrieved documents. The body of QA research most closely related to our work focuses on the framework of representing types of questions and automatic determination of question types from the given question. Ravichandran and Hovy [2002] proposed a question classification method that does not rely on external semantic knowledge, but rather classifies a question to different sets of

surface patterns, e.g. *ENTITY was born in ANSWER*, which requires *ENTITY* as an anchor phrase from the given question and impose no constraint on the semantic type of *ANSWER*. In contrast, we use a sizable set of question and answer pairs to learn how to classify a given question into a small number of types from the broad semantic types in the existing lexical knowledge base of WordNet.

In a study more closely related to our work, Ciaramita and Johnson [2003] used WordNet for tagging out-of-vocabulary term with supersense for question answering and other tasks. They discovered it is necessary to augment WordNet by employing complex inferences involving world knowledge. We propose a similar method *WikiSense*[1]that uses Wikipedia titles to automatically create a database and extend WordNet by adding new Wikipedia titles tagged with supersenses. Our method, which we will describe in the next section, uses a different machine learning strategy and contextual setting, under the same representational framework

Once the classes of the given questions have been determined, typical QA systems attempt to formulate and expand the query for each type of question or on a question by question basis. Kwok et al. [2001] proposed a method that matches the given question heuristically against a semi-automatic constructed set of question types in order to transform the question to effectively queries, and then extract potential answers from retrieved documents. Agichtein, Lawrence, and Gravano [2004] used question phrases (e.g., "*what is a*" in the question "*What is a hard disk?*") to represent the question types and learn query expansion rules for each question type. Prager et al. [2002] describe an automatic method for identifying semantic type of expected answers. In general, query expansion is effective in bringing more relevant document to the top-ranked list. However, the contribution to the overall question answering task might be marginal only. In contrast to the previous work, we do not use question types to expand queries, but rather use question types to filter and re-rank potential answers, which may contribute more directly to the performance of question answering.

Indeed, effective explicit question classification is crucial for pinpointing and ranking answers in the final stage of answer extraction. Ravichandran and Hovy [2002] proposed a method for learning untyped, anchored surface patterns in order to extract and rank answers for a given question type. However, as they pointed out, without external semantic information, surface classification suffers from extracting answer of improper class. Example shows a *where-is* question (e.g. *Where is Rocky Mountains?*) may be classified to the pattern "*ENTITY in ANSWER*" ("*Rocky Mountains in ANSWER*"), but with the retrieved text "...took photos of Rocky Mountains in the background when visiting...", the system may mistakenly identifies "*background*" as the answer. Intuitively, by imposing a semantic type of *LOCATION* on answers, we can filter out such noise (*background belongs* to the type of *COGNITION* according to WordNet). In contrast, we do not rely on anchor phrases to extract answers but rather use question types and redundancy to filter potential answers.

Another effective approach to extract and rank answers is based on redundancy. Brill, Lin, Banko, Dumais and Ng [2001] proposed a method that uses redundancy in two ways. First, relevant relation patterns (linguistic formulations) are identified in the retrieved documents, redundancies are counted. Second, answer redundancy is used to extract relevant

[1] The data of WikiSense will be made available to the public in the near future

answers. Distance between answer candidates and query terms are also considered in the proposed method through re-weighting. In our QA system, we use a similar approach of answer redundancy as our base line.

In contrast to the previous research in question classification for QA systems, we present a system that automatically learns to assign multiple types to a given questions, with the goal of maximizing the probability of extracting answers to the given question. We exploit the inherent regularity of questions and more or less unambiguous answer in the training data and use semantic information in WordNet augmented with rich named entities from Wikipedia.

3. Proposed Methods

In this section, we describe the proposed method for supersense tagging of Wikipedia article titles, minimally supervised question classification, and a simple redundancy based QA system for evaluation.

3.1 Problem Statement and Datasets

We focus on deploying question classification to develop an open domain, general-purpose QA system. Wikipedia titles, Wikipedia categories and YAGO are used in the process of generating WikiSense. For question classification, the 25 lexicographer's files in WordNet (supersenses) are used as the targeting class tagset. Both WordNet and WikiSense are used to generate the training data for classifying questions.

person	cognition	time	event	feeling
communication	possession	attribute	quantity	shape
artifact	location	object	motive	plant
act	substance	process	animal	relation
food	state	phenomenon	body	group

Table 1. The 25 lexicographer's files in WordNet, or supersenses.

At run time, we continue to use both WikiSense and WordNet for answer candidates filtering. Either the Web is used as the corpus, and Google is used as the information retrieval engine.

```
1)   Generate Large Semantic Category from Wikipedia titles (WikiSense)
         (Section 3.2.1)
2)   Training of Question Classifier using WikiSense and WordNet
         (Section 3.2.2)
3)   Redundancy QA System with Question Classification
         (Section 3.3)
```

Fig 1. Out line of the proposed method for QA system construction

3.2 Training Stage

The training stage of the proposed QA system consists of two main steps: generation of large scale, semantic category using Wikipedia (WikiSense) and training of fine-grained question classifier using WikiSense and WordNet. Figure 1 shows the steps of our training process and QA system.

3.2.1 Automatic Generation of Large Scale Semantic Category from Wikipedia

In the first stage of the training process (Step (1) in Figure 1), we generate a large scale, finer-grained supersense semantic database from Wikipedia. Wikipedia currently consists of over 2,900,000 articles. Every article in Wikipedia is hand tagged by volunteers with up to a few dozens of categories. There are 363,614 different categories in Wikipedia, some used in many articles, while many are used in only a handful of articles. These categories are a mixed bag of subject areas, attributes, hypernyms, and editorial notes. In order to utilize the information provided in Wikipedia categories, Suchanek, Kasneci, and Weikum [2007] developed YAGO as an ontology with links from Wikipedia categories to WordNet senses, thereby resolving the ambiguities that exist in category terms (e.g., *Capitals in Asia* is related to *capital city*, while *Venture Capital* is related to *fund*).

Although YAGO only covered 50% (182,945) of the Wikipedia categories, these categories covers of substantial part of Wikipedia articles. By using this characteristic in combination with YAGO, we use voting to heuristically determines which of the 25 WordNet lexicographer files the titles belongs to. Figure 2 shows the algorithm for categorizing Wikipedia titles using its Wikipedia categories and YAGO.

```
procedure WikiSense(Wikipedia, YAGO, WordNet)

      Declare Tags as list
      Declare Results as list

      for each Article in Wikipedia:
           Title := title of Article
(1)        Initialize Vote as an empty dictionary
           for each Category in Article:
(2)             if Category is supported by YAGO:
(3a)                 WordNetSense = YAGO(Category)
                     Append WordNetSense to Tags
(3b)                 WordNetSuperSense = WordNet(WordNetSense)
(4)                  Vote[WordNetSuperSense]++

           Class := superSense with most votes in Vote
(5)        append <Title, Class, Tags> to Results
(6)   return Results
```

Fig 2. Generation of WikiSense using Wikipedia titles/categories and YAGO

For every articles in Wikipedia, we use a dictionary to keep track of which supersense has the highest votes (Step (1)). In Step (2), all the category in the article are checked if they are supported by YAGO. Supported categories are than transformed to WordNet senses through YAGO in Step (3a). The transformed senses are than transformed again by WordNet to its corresponding supersense in Step Step (3b), and the supersense is voted once (Step (4)). Once all categories has been checked, title and its supersense with the highest votes is recorded, we also recored all the transformed WordNet senses for future uses (Step (5)). After all the articles in Wikipedia are processed, all the recorded results are returned in Step (6). In the entire process, WordNet is only used to transform a word sense to its supersense (lexical file).

We show the classification process and results of three example titles in Wikipedia in Table 2. None of these titles are in the WordNet vocabulary.

Wiki Title	Zenith Electronics
Categories	Consumer_electronics_brands, Electronics_companies_of_the_United_States, Companies_based_in_Lake_County_Illinois, Amateur_radio_companies, Companies_established_in_1918, Goods_manufactured_in_the_United_States
Senses	company#1 (3), electronics_company#1 (1), good#1 (1), 1:trade_name#1 (1)
Supersense	**noun.group (4)**, noun.attribute (1), noun.communication (1)
Wiki Title	Paul Jorion
Categories	Consciousness_researchers_and_theorists, Artificial_intelligence_researchers, Belgian_writers, Belgian_sociologists, Belgian_academics
Senses	research_worker#1 (2), writer#1 (1), sociologist#1 (1), academician#3 (1)
Supersense	**noun.person (5)**

Wiki Title	Hsinchu
Categories	Cities in Taiwan
Senses	city#1 (1)
Supersense	**noun.location (1)**

Table 2. Example of Wikipedia titles classification for generating WikiSense

3.2.2 Minimally Supervised Question Classification

In the second and final stage in the training process (Step (2) in Figure 1), we use WordNet and the previously introduced WikiSense to automatically create training data. Figure 3 shows the training algorithm for constructing question classification method. We use the Maximum Entropy Model to construct a single classifier with multiple outcomes (Step (1)). The input of this stage includes a semantic database to determine the outcomes and a set of question/answer pairs. For each question/answer pairs, we first determine whether the answer is listed in the input semantic database, unsupported question/answer pairs are neglected (Step (2)). In Step (3), a listed answer is transform into its supersense using semantic database as outcome(Step (3a)), features are extracted from question (Step (3b)). Finally, extracted features and transformed outcome is used as an event to train the classifier in Step (4). After all the listed question/answer pairs has been processed, the trained classifier is returned.

```
procedure QC Train(SemanticCategory, QASet)

(1)         Declare Classifier as Maximum Entropy Model

            for each <Q, A> in QASet:
(2)                   if A is not supported by SemanticCategory:
                          continue
(3a)                  Outcome  := SemanticCategory(A)
(3b)                  Features := ExtractFeatures(Q)
(4)                   Classifier.AddEvent(Features, Outcome)

            Classifier.Train()
(5)         return Classifier
```

Fig 3. Minimally Supervised training method of question classifier.

Most of the concepts in WordNet are basic vocabularies. Only few name entities can be found in WordNet, whereas Wikipedia contains a large amount of NEs. For instance NEs like “Charles Dickens” (writer) is in both WikiSense and WordNet vocabulary, while “Elton John” (singer), “Brothers in Arms” (song) or “Ben Nevis” (mountain) can only be found in WikiSense. However, WordNet, being handcrafted, still have much higher accuracy on basic words and phrases. Therefore we use both WikiSense and WordNet to cover common nouns as well as NEs.

There are three main features used in the training stage: (1) the supersense of NEs

found in the given question (2) the question phrase of the given question (3) any words in the given question.

Question	Named Entity Class	QuestionPhrase
In kilometres, how long is the Suez Canal?	noun.artifact	how-long
The action in the film "A View To A Kill" features which bridge?	noun.communication	which-bridge
Which famous authour was married to Anne Hathaway?	noun.person	which-author

Table 3. Example questions and features

At runtime, classification outcomes with probability higher than a threshold are retrieved. The value of the thresholds are set to a number of multiples uniform-distribution probability. In Section 4, we show experimental results of the proposed methods performed at different threshold.

3.3 Redundancy Based Question Answering System

We use Google as our document retrieval engine to search the entire Web. Only the snippets of the top 64 retrieval results are used. After retrieving snippet passages, we take advantage of the large amount of retrieved text to extract candidate and rely on redundancy to produce the answer. Previous work shows that answer redundancy is an effective technique for the QA task (Brill et al. [2001]).

Once answer candidates are extracted and redundancy counted, candidates are re-ranked based on question classification results. We retain and make use of several predicted question types (with probability higher than a threshold), in other words, the given question may be classified to multiple classes. This is reasonable due to the characteristic of our class tagset. Consider the question "*Where were Prince Charles and Princess Diana married?*". It may be answered with either name of a city (*London),* or name of a church (*St Paul's Cathedral)*, therefore the question type could be either LOCATION or ARTIFACT. After the passages are retrieved, answer candidates are extracted and classified using WordNet and WikiSense. Finally, we re-rank the 20 most frequent candidates by order the candidates in descending order of question type probability, and then by frequency counts. Finally, we produce the top n candidates as ouput.

4. Experimental Results Evaluation

In this section r, we describe experimental settings and evaluation results. In Section 4.1, we describe in detail the experimental settings and evaluation matrices. Then evaluation results and analysis of WikiSense and question classification are discussed in Section 4.2 and Section 4.3. Finally, we report the performance of the classifier on a simple redundancy based QA system and evaluate its effectiveness in Section 4.4.

4.1 Experimental Setting and Evaluation Matrices

In the first experiment, we explain and analysis the result and coverage of WikiSense, which is then used in the second experiment to classify questions in addition to WordNet.

We collected 5,676 question/answer pairs as the training data from the Quiz-zone Web site (http://www.quiz-zone.co.uk/), an online quiz service with popular culture and general knowledge questions designed to be answered by human. To evaluate our method, one tenth of the question/answer pairs is separated from the training data as the evaluation data. Correct classes of the questions are labeled by human judges in order to evaluate the performance of question classification.

We then used the proposed minimally supervised training method to generate two question classifiers based on different database setting. In the first experiment, we only used WordNet to generate data to train the first classifier (baseline), and then compared the classifier with the second classier trained on both WordNet and WikiSense. The purpose is to show the amount of improvement contributed by WikiSense, if any. Since WordNet is constructed by human, we consider it to have higher precision. Therefore, WordNet is used when conflicting arises between WikiSense and WordNet. The results of both classifier are presented and compared in term of recall and precision rates.

4.2 WikiSense

An implementation of the proposed method classifies about 55% of all titles in Wikipedia, resulting a large scale, finer-grained, supersense semantic category containing 1,581,865 entities.

Unclassified titles are usually caused by articles with little or no categories so their semantic type can not be accurately determined. However, the result does not imply the classification method has low coverage. Unlike most offline encyclopedias, Wikipedia is an ongoing collaborative work. Thousands of new and unfinished articles are created by volunteers or robots daily. The Wikipedia editorial principle state that every Wikipedia article should belong to at least one category, therefore uncategorized titles usually belongs to articles still in the early stage of development (called “stubs” in the Wikipedia community).

4.3 Question Classification

In this section, we report the evaluation results on using the trained classifier to classify questions. Figure 4 shows the results of the two classifiers in terms of recall and retrieval size at different level of threshold (in multiples of 0.04, the average probability). At same recall performance, the lower retrieval size results in higher precision. As Figure 5 shows, higher precision is achieved with higher threshold, trading off recall. Notice that the recall of both classifiers gradually decreases when threshold increases from one to five times of uniform probability. Above threshold 5, recall of both classifiers decreases rapidly. Considering recall being crucial to question classification task in order to prevent early elimination of the correct answer candidates, we focus our analysis on thresholds lower than 5. We can see that the

precision increases for both classifiers as threshold increases. The combined classifier was able to achieve slightly higher recall and higher precision of 9% at threshold of 2 times of uniform probability.

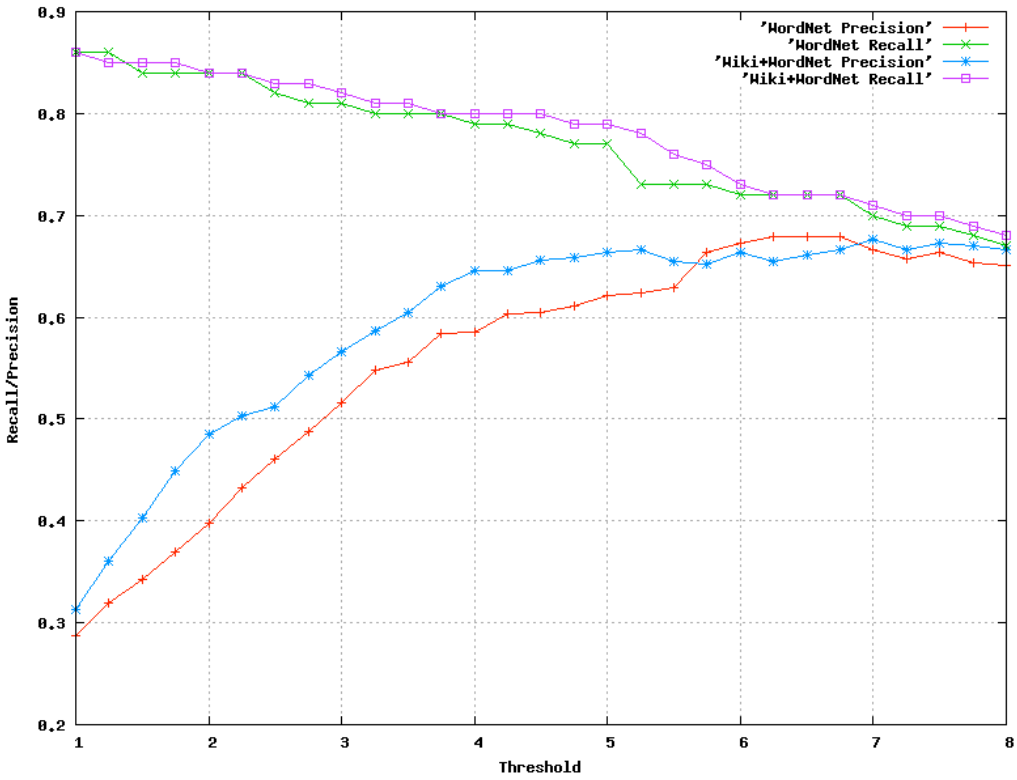

Fig 5. Performance in terms of precision and recall at different threshold.

4.4 Question Answering

In this experiment, we first run our QA system with out any question classification as our baseline. We then run the same system on the same evaluation dataset using two different question classifier, one trained by WordNet and the other trained on WordNet plus WikiSense.

Threshold	Top 1	MRR
Baseline	34%	0.451
1.0	39%	0.476
1.5	40%	0.482
2.0	42%	0.501
2.25	42%	0.503
2.5	35%	0.457

(a) WordNet

Threshold	Top 1	MRR
Baseline	34%	0.451
1.0	43%	0.492
1.5	43%	0.503
2.0	44%	0.509
2.25	43%	0.512
2.5	35%	0.457

(b) WikiSense + WordNet

Table 4. Top 1 precision and MRR result of deploying the 2 classifiers

Table 4 lists Top 1 precision and MRR of our baseline system and the system with the two classifiers at varying thresholds. As we can see, by including question classification, both systems performed better than baseline. With the enhancement of WikiSense, results in Table 4(b) achieve significantly higher MRR and top 1 precision comparing to system with a classifier trained on WordNet only (see Table 4(a)). The best performance of both MRR and top 1 precision was achieved by the system with both WikiSense and WordNet. At threshold of 2.25, the MRR was higher than the baseline by 0.061, and top 1 precision is higher by 9%.

5. Conclusions

Many future research directions present themselves. For example expanding the coverage of WikiSense using other characteristics of Wikipedia, such as internal link structure, article contents, information boxes and Wikipedia templates, minimally supervised training for automatically supersense tagging on Wikipedia title, and a more complex QA system that take full advantage of finer-grained classification.

In summary, we have introduced a method of minimally supervised training for fine-grained question classification using an automatically generated supersense category (WikiSense) and WordNet. The method involves supersense tagging of answers to generate training data, and using Maximum Entropy model to build question classifiers. We have implemented and evaluated the proposed methods using a simple redundancy based QA system. The results show the method substantially outperforms the baseline of now using question classification.

References

[1] E. Agichtein and S. Lawrence and L. Gravano, Learning to Find Answers to Questions on the Web, ACM Transactions on Internet Technology (TOIT), volume 4, pp. 129-162, 2004

[2] E. Brill and J. Lin and M. Banko and S. Dumais and A. Ng, Data-Intensive Question Answering, In Proceedings of the Tenth Text REtrieval Conference (TREC), pp. 393-400, 2001

[3] M. Ciaramita and M. Johnson, Supersense Tagging of Unknown Nouns in WordNet, Conference on Empirical Methods on Natural Language Processing (EMNLP), pp. 168-175, 2003

[4] C. Fellbaum, Wordnet: An Electronic Lexical Database, ISBN: 026206197X, May 15, 1998

[5] C. Kwok and O. Etzioni and D. S. Weld, Scaling question answering to the web, ACM Transactions on Information Systems (TOIS), Volume 19, Issue 3, pp. 242-262, 2001

[6] MetaWeb Technologies, Freebase Wikipedia Extraction (WEX) version June 16, 2009, http://download.freebase.com/wex/ , 2009

[7] J. Prager and J. Chu-Carroll and K. Czuba, Statistical answer-type identification in open-domain question answering, Proceedings of the second international conference on Human Language Technology Research, pp. 150-156, 2002

[8] F. M. Suchanek and G. Kasneci and G. Weikum, Yago - A Core of Semantic Knowledge, 16th international World Wide Web conference (WWW), 2007

[9] D. Ravichandran and E. Hovy, Learning Surface Text Patterns for a Question Answering System, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 41-47, July 2002