Integrating Knowledge Resources and Shallow Language Processing for Question Classification

Maheen Bakhtyar
Department of CSIM,
Asian Institute of Technology, Thailand.
Department of CS&IT,
University of Balochistan, Pakistan.
Maheen.Bakhtyar@ait.asia

Asanee Kawtrakul
Department of Computer Engineering,
Kasetsart University,Thailand.
NECTEC, Pathumthani,
Thailand.
asanee_naist@yahoo.com

Abstract

Typically, Question Classification (QC) is the first phase in Question Answering (QA) systems. This phase is responsible for finding out the type of the expected answer by having the answer space reduced by pruning out the extra information that is not relevant for the answer extraction. This paper focuses on some *Location* based questions and some *Entity* type questions. Almost all the previous QC algorithms evaluated their work by using the classes defined by Li and Roth (2002). The coarse grained classes *Location* and *Entity* both have fine grained class *Other*. In this paper we target and present the mechanism to create new classes to replace the *Other* classes in *Location* and *Entity* class. Additionally, we also present an automatic hierarchy creation method to add new class nodes using the knowledge resources and shallow language processing. We also show how language processing and knowledge resources are important in the question processing and its advantage on Answer Extraction phase.

1 Introduction

Usually people are interested in the exact answer and do not desire to look for the answer themselves in long list of documents. Exact answer is more interesting and useful than getting a list of documents.

Query analysis, processing or classification phase have been always emphasized. The following examples [1] show the importance of this phase with respect to the Answer Extraction.

Example 1: Who was the first American to walk in space?. The answer sentence obtained is *"In 1965 astronaut Edward White became the first American to "walk" in space during the flight of Gemini 4"*[2]. Suppose the question is classified as *Human:Individual* by some classification mechanism. We notice that the answer line contains the matching string "first American to walk in space" therefore, the answer to the question is to be selected from the remaining part "1965", "Edward White" or "Gemini 4". Correct classification now leads us to the answer *Edward White*.

Example 2: What day and month did John Lennon die?. If this question is classified as *Number:Date*, it means that only date type will be targeted from the text. This implies that the question when correctly classified will give a hint about the answer which helps the system in judging and extracting the answer from the corpus.

The questions can be categorized mainly in two ways i.e. considering the question word and second the answer type. Ray et al. (2010) categorizes the factoid questions first in the categories such as *"who", "why", "what", "where", "how" and "when"* and classify them based on the two level hierarchy of classes defined by Li and Roth (2002) and shown in Table 1.

2 Problem Statements

Question Classification is important and helpful for extracting the answers. A correct and meaningful classification will lead the system to more efficient and correct answer extraction mechanisms. On the other hand, a wrong or meaningless classification will not improve the answer extraction and might become a cause of inaccurate final results.

[1] Questions and answer sentence taken from TREC-10 Text-REtrieval-Conference-10 (2001)

[2] This line is taken from the document number *DOCNO: AP890527-0145* and contains the answer to this question

Proceedings of the KRAQ11 Workshop, pages 22–28,
Chiang Mai, Thailand, November 12, 2011.

Table 1: Coarse and Fine grained classes

Coarse	Fine
ABBR	abbreviation, expansion
DESC	definition, description, manner, reason
ENTY	**animal, body, color, creation, currency, disease/medical, event, food, instrument, language, letter, *other*,plant, product, religion, sport, substance, symbol, technique, term, vehicle, word**
HUM	description, group, individual, title
LOC	**city, country, mountain, *other*, state**
NUM	code, count, date, distance, money, order, other, percent, period, speed, temperature, size, weight

2.1 Insufficient classes in the taxonomy

Question classes defined and labeled in UIUC[3] dataset by Li and Roth (2002) are most widely used in the previous work (Quan et al. (2011), Song et al. (2011), Yu et al. (2010), Buscaldi et al. (2010), Huang et al. (2007) and Boldrini et al. (2009)). Many of the researchers developed their systems using these classes and the labeled question dataset. In the labeled dataset, if a question is not mapped to some class, it is placed into the fine grained class *Other*. Assigning to a class *Other* is not very helpful in the answer extraction. For example, in case of *Location* category, *Location:Other* will only prune out *city, country, mountain* and *state* as possible answer categories. Therefore, a close analysis of questions belonging to this class is needed and a new set of classes is required to overcome this deficiency.

We currently focus on two of the coarse grained classes; *Location* and *Entity*; and all their fine grained classes. It is also observed that many of the fine grained classes are missing in the existing class hierarchy which needs to be mapped to the questions. For instance, the class *river*, *lake* or any other *water body* is not present in the existing class taxonomy whereas some questions require such classes e.g. the question *What body of water are the Canary Islands in ?* is currently placed in class *LOC:Other* by Li and Roth (2002). This assigned class neither gives an exact hint nor helps to filter the candidate answers. Whereas, mapping it to a class such as *waterbody* makes it more meaningful and easier to find the answers. Similarly, the question *"what is Bill Gates of Microsoft e-mail address"* ? is labeled as *LOC:Other* by the authors. If this question is searched using a search engine, a lot of documents will be returned having all the key concepts in the question. A chunk of text containing the answer is as follows, *"All the Good Emails get sent to another Bill Gates Email Address, which he checks twice a week. Because He knows everyone will be looking for his email address under @microsoft.com. The Employees who checks his email under billg@microsoft.com send it to the one he checks"* [4]. This chunk from the document contains all the question keywords. Without the classes defined, we do not know which part of the chunk is more important. Whereas, if we determine that the answer should be an email address, then we only need to target the email addresses in the text without taking care of the rest of the document. Therefore, the detail of classes and subclasses is needed to cover more and more questions instead of assigning them to the *LOC:Other* class.

Li and Roth (2002) show that among 500 questions in TREC 10, 62% of the location questions belong to the class *Other*. The highest number of questions lie under the location category *Other* which is actually not very helpful or meaningful in extracting the answer. It means that about 62% of the location questions will be answered during the answer extraction phase without making use of the classes, despite the efforts put into classification phase. Similarly, 13% of the entity questions belong to the class *Other*. *Entity* class has 22 fine grained classes and the large number of questions are mapped to *Other* after *animal* and *substance*. Later, Li and Roth (2006) again gave a statistics of distribution of questions in each class of TREC 10 and 11 Text-REtrieval-Conference (1999 to 2007) questions, collectively. They observed that out of 1000 questions, 195(19.5%) are *Location* based. In *Location* based questions, there are 22.6% questions mapped to class *city*, 10.8% questions about class *country*, 2.6% about *mountain*, 58.5% are mapped to class *other*, and 5.6% questions are mapped to class *state*.

One of the main advantage of replacing the class *Other* with fine grained classes is that it makes assignment of a single question to multiple classes/-

[3] http://cogcomp.cs.illinois.edu/Data/QA/QC/

[4] http://email.about.com/b/2009/05/30/how-can-i-email-bill-gates-what-is-bill-gatess-email-address.htm

subclasses more efficient and effective. A question may implicitly belong to all the subclasses thus, it increases the class coverage.

2.2 Unavailability of automatic class creation mechanism in the hierarchy

In the previous section we show that the new hierarchy of classes and subclasses is needed and effective for efficient answer extraction. Creating new classes manually for each and every possible question is impossible and we need an automatic mechanism to create and assign new classes. Li and Roth (2002) presented a two level hierarchy with a fixed number of classes. Whereas a more general method to create and assign new classes to the questions is required. The new classes may be organized in any number of levels in the hierarchy and can be assigned accordingly.

Our target is to fill the gap of the unavailable classes. We propose a technique that automatically creates new classes and classify the questions having *"what |which NP ... "* pattern. Our technique is based on the language processing and external knowledge resources.

3 Methodology

In this section we propose a methodology for creating the hierarchical structure that represents the classes, and the mechanism to automatically add new classes into the hierarchy.

3.1 Classes in form of a hierarchy

We propose an algorithm that creates hierarchical class taxonomy by placing the existing classes into appropriate position in the tree, and add new classes which are missing in the previous taxonomy, as shown in Figure 1. The question *"Which is the largest island in Thailand?"* is previously mapped to the class *LOC:Other* because there is not any appropriate class available. The *LOC:Other* class does not help much to extract the answer from the given text chunk *"Phuket is now Thailand's most important tourist destination, offering a variety of beaches, attractions and exciting night life. Koh Phuket is Thailand's largest Island. It is 50 km long north to south and 21 km wide and joined to the mainland by Sarasin bridge.Phuket has been inhabited since the early days of mankind by ancient tribes and this still keeps archaeologists occupied to find out the history from the early days.*[5]*"*. If the same question is mapped to the class *LOC:PHY:Natural:Island* or even if to the class *LOC:PHY:Natural*, it will help to locate the *natural locations* or *islands* inside Thailand from the given text chunk.

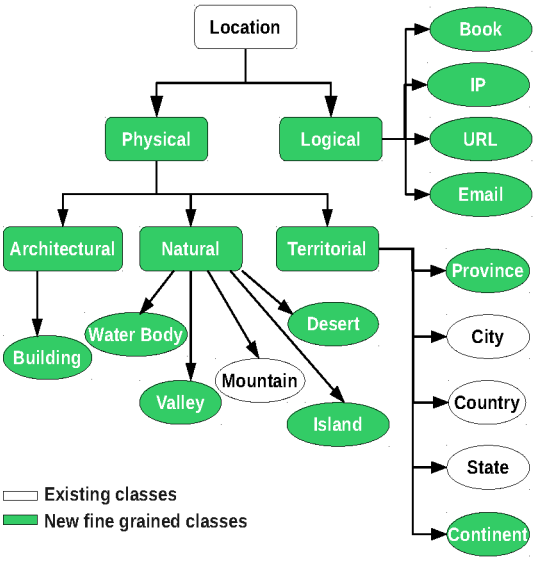

Figure 1: Location Class Hierarchy

Similarly, the question mentioned in the previous section, *"What is Bill Gates of Microsoft e-mail address"* ?, if mapped to subclass *LOC:Logi:Email* or to class *LOC:Logi* will give the hint that some logical location such as URL, email (note that they have fixed patterns and can be extracted easily from the text chunk) is required as an answer. The two classes *Physical* and *Logical* are created by hand and later the subclasses can be inserted.

3.2 Automatic class creation in the hierarchy

In this problem, we will target the questions having the pattern *"what |which NP ... "*. The questions starting with the *What* and *Which* question words, followed by a Noun Phrase (NP), have their expected answer type inside the NP. The expected answer type/question class will be the focus of the NP.

We examined the distribution of the target pattern questions over the existing class hierarchy. We used 1500 questions consisting of 1000 training questions available at UIUC, and 500 questions from TREC 10. We observed that 23% of the questions belong to class *Entity* and 16% belong to class *Location*. Out of these, 16% and 54% belong to classes *ENTY:Other* and *LOC:Other* re-

[5] http://www.beachpatong.com/

spectively. We also observed that 30% of the *ENTY:Other* and 54% of the *LOC:Other* questions are of our target pattern *"what |which NP ... "*. It shows that the proportion of question matching this pattern seems adequate to get started. Therefore, we will focus on the class *Entity:Other* and its subclasses as shown in Figure 2. The similar approach can be applied to the *LOC:Other* class with separate set of patterns e.g. the question *"which part of the university has most trees?"* is a *Location* question having no defined class in the initial hierarchy as well as in the newly created hierarchy shown in Figure 1. Now, the same idea can be applied to this class using different set of patterns, but to keep the initial work simple, we target the subset of question classes and question patterns. Once we have developed a system to add new nodes for this set of questions, we can define similar algorithms for the other set of questions.

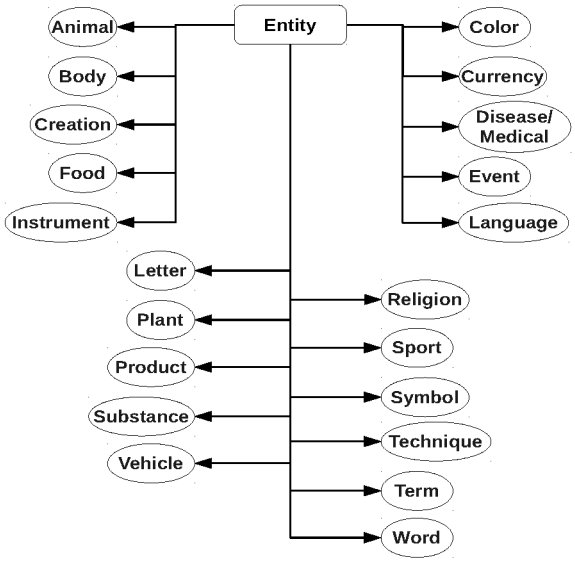

Figure 2: Entity Class Hierarchy

3.2.1 Noun Phrase and Head Noun

Noun phrases have a head noun surrounded by some modifiers such as possessives, adjectives. For example, *"Which **Thailand's island** has highest number of tourists?"* or *"Which **dark green plant** is beneficial to fight cancer? "*

After the first NP is identified in the question, the next task is to determine the head noun e.g. *island* and *plant* in the examples above. Head noun is the target focus of the question and also a candidate class to be added as a node in the hierarchy. For example, in the question *what four forms does gold occur in?* has NP *four forms* and the head noun in this NP is the *forms* which is a candidate new node in the hierarchy. Similarly, the question *"which fungi cause the skin infection?"* has NP and head noun *fungi* and is a candidate for the class in the classes hierarchy.

3.2.2 Adding a class based on similarity calculation and knowledge resources

After finding the focus of the NP i.e. the candidate class for the hierarchy, we cannot directly add the node in the hierarchy. Adding each and every candidate class directly into the hierarchy will make the hierarchy grow very rapidly. We need to consider the relationship between the candidate class and the existing classes before adding a new node. Therefore, first we calculate the similarity between the new candidate and the existing nodes in the hierarchy. If the similarity value between candidate class and some existing class is greater than a threshold t then that existing class is assigned to the candidate class, otherwise a new node is added. The basic framework is shown in Figure 3.

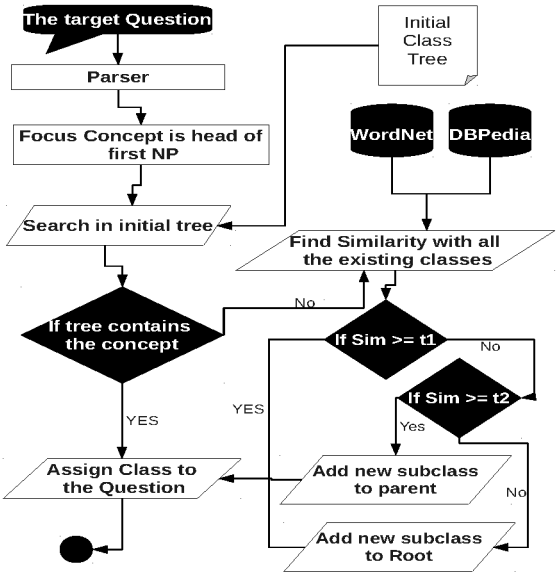

Figure 3: Entity Classification Framework

For calculating the similarity, we use two knowledge resources; WordNet Miller (1995) and DBPedia[6]. DBPedia is the structured version of Wikipedia and one of the largest structured data available online. We use two knowledge bases to cross check and support the answers from both the resources. After calculating the similarity based on WordNet and DBPedia, an average similarity value is used to compare with the threshold val-

[6]http://wiki.dbpedia.org/

ues. We take two threshold values t_1 to classify the question using existing classes and t_2 to add new node as a sub-class of some existing class, where $t_1 > t_2$. If the similarity value is less than both the threshold values then the new node is created but as a child of the root node. The basic algorithm is shown in Algorithm 1. In the algorithm, *AssignClass(Q, some_class)* classifies the question Q as *some_class*. *InsertChildToParent(some_child,some_parent)* creates the class *some_child* as a child of class *some_parent*.

Algorithm 1 Classification

Require: A natural language question Q
Require: Threshold values t_1 and t_2
NP := First Noun Phrase after the Question Word
candidate := Extracted Head Noun from NP
root := Root of the tree
n := Number of tree nodes
for $i = 1$ to n **do**
  SimWN :=$\text{Sim}_{\text{OSS}}(node[i]$, candidate) using WordNet
  SimDB :=$\text{Sim}_{\text{OSS}}(node[i]$, candidate) using DBPedia
  similarity := (SimWN + SimDB)/2
  if similarity$\geq t_1$ **then**
    AssignClass(Q,node[i])
    BREAK LOOP
  end if
  if similarity$\geq t_2$ **then**
    InsertChildToParent(candidate,[node[i])
    AssignClass(Q,candidate)
    BREAK LOOP
  end if
  InsertChildToParent(candidate, root)
  AssignClass(Q,candidate)
  BREAK LOOP
end for

To find the similarity, we use the algorithm OSS by Schickel-Zuber and Faltings (2007), they reported that their results are better than the existing algorithms. They provide a mechanism to find the semantic relatedness of two concepts.

A high value of Similarity function (Sim_{OSS}) means the concepts are highly related. This value depends on the distance between one concept to another in the Ontology; WordNet and DBPedia in our case.

There is a relationship between the similarity value and the size of the hierarchy. If similarity of concepts is low, it compels to add new nodes into the tree. This means the size of the tree will depend on the threshold set for the addition of new nodes. If the threshold value is big, then tree size will increase because most of the new candidate classes will be added as new node. Therefore, a reasonable values of t_1 and t_2 are to be determined for a reasonable number of nodes in the tree. We have set the threshold values manually in our framework, $t_1 = 0.7$ and $t_2 = 0.5$. If the similarity of candidate class with any of the existing classes is greater than t_1, then the question is mapped to that existing class and no new node is added in the tree.

As we know that in the question *"which fungi cause the skin infection?"*, the head noun is *Fungi*. Now, to decide whether this node should be added or not, we find the similarity with the nodes in the existing hierarchy. Similarity calculation for some concepts is shown in Table 2. A high similarity is observed between the concepts *fungus* and *plant*. Therefore, the question will be classified as

Table 2: Similarity calculation

	Sim WN	Sim DBP	Avg
Plant-Fungus	0.86	0.7	0.78
Disease-Artery	0.96	0.0	0.48
Disease-Disease	1.0	1.0	1.0
Musical-Event	0.74	0.0	0.37

the type *Plant*.

We used more than one ontologies, as there might be some relationship missing in either of ontologies. Therefore, to cross check the relationship more than one ontologies are used. For example, in case of DBPedia the similarity value is 0 for *Disease-Artery* and *Musical-Event*, whereas WN gives high similarity.

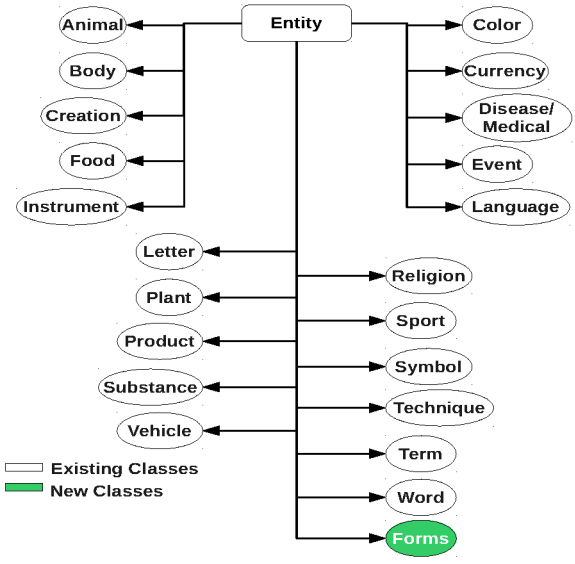

Figure 4: Adding new node

Another question *"what four forms does gold occur in?"* has the target focus (head noun) *forms*. The similarity computed is less than both the threshold values for all the nodes therefore it is added into the hierarchy as a child of the root node.

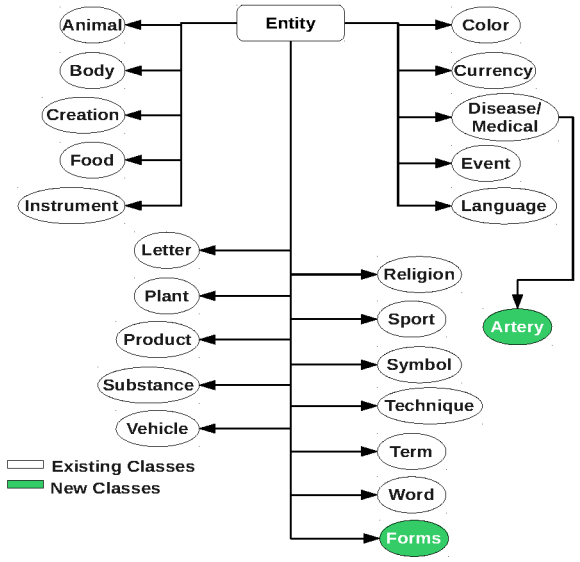

Figure 5: Adding new child

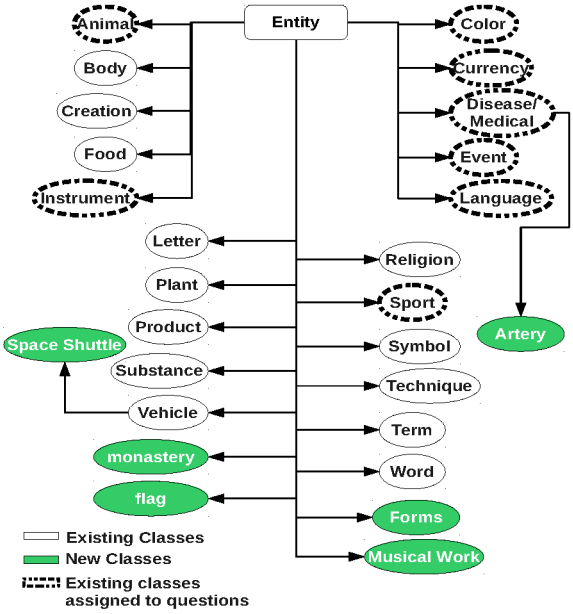

Figure 6: Chunk of new hierarchy

The resulting hierarchy is shown in Figure 4.

For the third case, take the question "what artery is responsible for taking blood from heart to the lungs? ". Assume that after finding similarity between head noun *artery* and all the nodes, none is greater than the threshold t_1. We computed the average similarity of the the head noun with *Disease* as shown in the Table 2. Similarity came out to be $0.48 \approx 0.5$, which is same as t_2, therefore, the new node will be created as a child as shown in Figure 5.

4 Experiments and Discussion

We performed the experiments on set of approximately 20 questions of the pattern *"what |which NP..."* selected from UIUC dataset and some created by hand. To test the system, we first obtain the focus (head noun of NP) of the questions and then check the similarity based on the steps defined in Algorithm 1. Using the rules in the algorithm we populated the hierarchy and assign the classes to the questions. A visual chunk of the resulting tree is shown in Figure 6 and some of the questions are as follows:

Example of questions classified using existing classes are *what gaming devices were dubbed Mississippi marbles and Memphis dominoes?* (mapped to instrument) and *what meter was invented by C.C. Magee in 1935 ?* (mapped to instrument).

We compare the hierarchy built using our approach (Figure 6) with the course and fine grained classes defined by (Li and Roth, 2002) shown in Table 1. We examine some questions and show how the the new classes in the hierarchy can be more helpful to extract the answer.

The question *"What monastery was raided by Vikings in the late eighth century ?"* is currently mapped to *ENTY:Other* whereas using our hierarchy it is mapped to the class *monastery*. If any answer extraction module uses this class instead of *ENTY:Other*, it will skip extra information and will only look for names or information about some monastery. Another example is the question *"which space shuttle was first launced by NASA?"* can be mapped to class *vehicle* which is also informative but our algorithm puts it in the class *Space Shuttle* which makes it more informative. The answer extraction module of any system will now search for the Space Shuttle and its related information.

The initial taxonomy for type *Entity* contained 21 fine grained classes. Our hierarchy, after performing our this initial experiments, had total of 27 classes. It means 22% of the hierarchy consists of new classes. 33% of the newly created classes are added as the sub-classes of some existing class and the remaining are added as direct child of the root.

Our main focus is to develop a more informative class hierarchy. The hierarchy contains more informative fine grained classes which will help the answer extraction phase to locate the answer more precisely. We do not present a complete classi-

fication scheme but initially only for the specific pattern of questions as discussed earlier.

Answer extraction phase requires the question to be classified in some manner. If a classification mechanism is developed by using our set of classes, then answer extraction technique be more helpful to extract the answer.

5 Conclusion and Future Work

We propose a new hierarchy for the questions that earlier belonging to the class *Location:Other* or *Entity:Other* . We show that classifying the questions into *"Other"* is not very useful for the answer extraction phase. These two classes are now represented as a hierarchy which is populated using some NLP techniques and knowledge resources i.e. WordNet and DBPedia. We also analyzed how the new hierarchy helped to prune out the extra unnecessary details for efficient answer extraction.

This is the initial work carried out with extremely limited questions. We only focused on the question with a specific pattern for generating the new hierarchy using knowledge resources. We plan to work on the remaining question types and patterns in the future. Moreover, we also plan to target the other coarse classes, "*NUM*" having sub-type "*Other*".
Additionally, we plan to label the questions and publish with the hierarchy obtained for all the questions set so a new set of classes is obtained and is comparable for the other researchers.

References

E. Boldrini, S. Ferrández, R. Izquierdo, D. Tomás, O. Ferrández, and J. L. Vicedo. 2009. A proposal of expected answer type and named entity annotation in a question answering context. In *Proceedings of the 2nd conference on Human System Interactions*, HSI'09, pages 315–319, Piscataway, NJ, USA. IEEE Press.

Davide Buscaldi, Paolo Rosso, José Manuel Gómez-Soriano, and Emilio Sanchis. 2010. Answering questions with an n-gram based passage retrieval engine. *J. Intell. Inf. Syst.*, 34:113–134, April.

Peng Huang, Jiajun Bu, Chun Chen, and Guang Qiu. 2007. An effective feature-weighting model for question classification. In *Proceedings of the 2007 International Conference on Computational Intelligence and Security*, CIS '07, pages 32–36, Washington, DC, USA. IEEE Computer Society.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Xin Li and Dan Roth. 2006. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(03):229–249.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.

Xiaojun Quan, Liu Wenyin, and Bite Qiu. 2011. Term weighting schemes for question categorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):1009–1021.

Santosh Kumar Ray, Shailendra Singh, and B.P. Joshi. 2010. A semantic approach for question classification using wordnet and wikipedia. *Pattern Recognition Letters*, 31(13):1935 – 1943. Meta-heuristic Intelligence Based Image Processing.

Vincent Schickel-Zuber and Boi Faltings. 2007. Oss: a semantic similarity function based on hierarchical ontologies. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 551–556, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Wanpeng Song, Liu Wenyin, Naijie Gu, Xiaojun Quan, and Tianyong Hao. 2011. Automatic categorization of questions for user-interactive question answering. *Inf. Process. Manage.*, 47:147–156, March.

Text-REtrieval-Conference-10. 2001. Trec-10 question answering data.

Text-REtrieval-Conference. 1999 to 2007. Trec qa main page.

Zhengtao Yu, Lei Su, Lina Li, Quan Zhao, Cunli Mao, and Jianyi Guo. 2010. Question classification based on co-training style semi-supervised learning. *Pattern Recogn. Lett.*, 31:1975–1980, October.