**LEARNING QUESTION CLASSIFIERS**

Ashwin kumar Muruganandam

Department of Computer Science

University of Texas at Dallas

[{axm164730}@utdallas.edu](mailto:%7baxm164730%7d@utdallas.edu)

**Abstract**

We try to replicate the work of “[Learning Question Classifiers](http://www.aclweb.org/anthology/C02-1150) ”by Xin Li and Dan Roth. In order to respond correctly to a free form factual question given a large collection of texts, one needs to understand the question to a level that allows determining some of the constraints the question imposes on a possible answer. These constraints may include a semantic classification of the sought after answer and may even suggest using different strategies when looking for and verifying a candidate answer. This paper presents a machine learning approach to question classification. We learn a hierarchical classifier that is guided by a layered semantic hierarchy of answer types, and eventually classifies questions into fine-grained classes. We show accurate results on a large collection of free-form questions used in TREC 10.

**Introduction**

Open ended question answer system is one of the domains in Natural Language Processing where amazing research works are taking place every day and evolving. For any Question answer system, the important or first goal should be identifying the question type. The question classification module in a question answering system has two main requirements. First, it provides constraints on the answer type that allows further pre-processing to precisely locate and verify the answer. Second, it provides information that down-stream processes may use in determining answer selection strategies that may be an answer type specific rather than uniform. Our goal is to categorize questions into different semantic classes that impose constraints on potential answers, so that they can be utilized in later stages of the question answering process. For example, when considering the question Q: What Canadian city has the largest population?, the hope is to classify this question as having answer type city, implying that only candidate answers that are cities need consideration.

**Related Work**

Open-domain question answering (Lehnert, 1986; Harabagiu et al., 2001; Light et al., 2001) and story comprehension (Hirschman et al., 1999) have become important directions in natural language processing. Question answering is a retrieval task more challenging than common search engine tasks because its purpose is to find an accurate and concise answer to a question rather than a relevant document. The difficulty is more acute in tasks such as story comprehension in which the target text is less likely to overlap with the text in the questions.

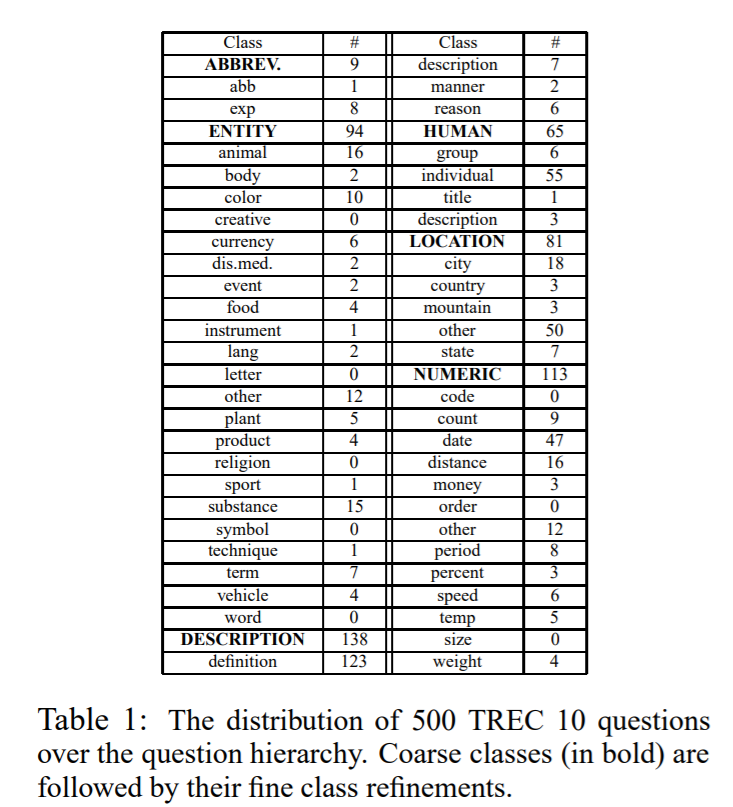
For this reason, advanced natural language techniques rather than simple key term extraction are needed. One of the important stages in this process is analyzing the question to a degree that allows determining the “type” of the sought after answer. In the TREC competition (Voorhees, 2000), participants are requested to build a system which, given a set of English questions, can automatically extract answers (a short phrase) of no more than 50 bytes from a 5-gigabyte document library.

Participants have re- Research supported by NSF grants IIS-9801638 and ITR IIS0085836 and an ONR MURI Award. alized that locating an answer accurately hinges on first filtering out a wide range of candidates (Hovy et al., 2001; Ittycheriah et al., 2001) based on some categorization of answer types. This work develops a machine learning approach to question classification (QC) (Harabagiu et al., 2001; Hermjakob, 2001). Our goal is to categorize questions into different semantic classes that impose constraints on potential answers, so that they can be utilized in later stages of the question answering process. For example, when considering the question Q: What Canadian city has the largest population?, the hope is to classify this question as having answer type city, implying that only candidate answers that are cities need consideration.

**Approach**

**Architecture**

We define a two-layered taxonomy, which represents a natural semantic classification for typical answers in the TREC task. The hierarchy contains 6 coarse classes (ABBREVIATION, ENTITY, DESCRIPTION, HUMAN, LOCATION and NUMERIC VALUE) and 50 fine classes, Table 1 shows the distribution of these classes in the 500 questions of TREC 10. Each coarse class contains a non-overlapping set of fine classes. The motivation behind adding a level of coarse classes is that of compatibility with previous work’s definitions, and comprehensibility

****

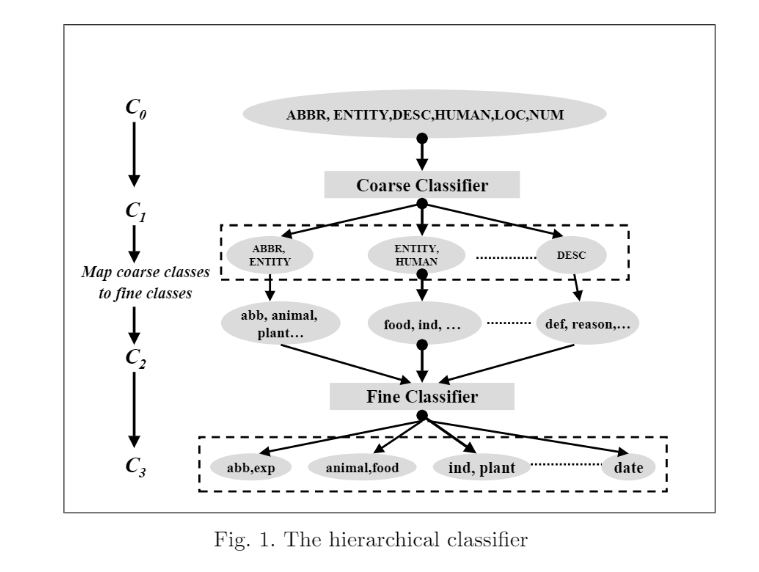
**Feature Space**

The primitive feature types extracted for each question include words, pos tags, chunks, named entities, head chunks (e.g., the first noun chunk in a sentence) and semantically related words (words that often occur with a specific question class).

Pos tags are extracted through nltk tool kit. Named entities through standford NER classifier, head chunks through practnlp tool kit and semantically related words from wordnet and nltk tool kit.

We construct our training data set by constructing a sparse matrix from these features.

Li and Roth use different ways to extract features. They haven’t implemented their own parser and semantic relation between words for that. Since they haven’t mentioned about the implementation details and since that’s not our scope (i.e) we care more about predicting question class label rather than implementing ways to extract feature space. We used the already available ones.



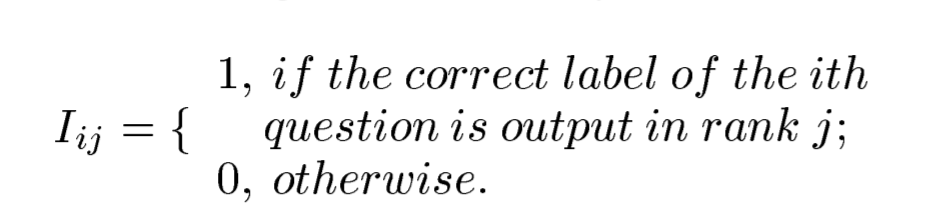
**Ambiguities**

One difficulty in the question classification task is that there is no completely clear boundary between classes. What is bipolar disorder? Whether it belong to definition or disease medicine. What do bats eat? Does it mean food, plant or animal? What is the PH scale? Is the person looking for a numeric value or a definition?

**Decision Model**

It is hard to categorize those questions into one single class and it is likely that mistakes will be introduced in the downstream process if we do so. To avoid this problem, we allow our classifiers to assign multiple class labels for a single question. This method is better than only allowing one label because we can apply all the classes in the later processing steps without any loss. We use naiveBayes and linear SVC to do prediction to do the prediction. Li and Roth user their SNOW model to the prediction. We can constructed our own Naïve Bayes library that outputs the probability of k top contenders, K <= 5. Our naïve bayes model is a basic one, we haven’t implemented state of art smoothing techniques, which are very important for our model because the number of features we have in our case is very high. Adding to that what we have is sparse matrix. So smoothing techniques should be very important since most of the data in the matrix are zero.

**Evaluation**

The paper evaluates on two components. Predicting only 1 and predicting top 5 classes. It claims that due to ambiguity, we need to provide some slack to the system. In our project we find predict only one. Not top 5 classes. For top 5 classes following is the expression to compute the accuracy. 

**Data Set**

The entire dataset is provided by Li and Roth in this website <http://cogcomp.org/Data/QA/QC/>. We trained our model with Training set 5 (5500 labelled question) and our test data set is Trec 10 Questions (500 labelled questions). Data are collected from four sources: 4,500 English questions published by USC (Hovy et al., 2001), about 500 manually constructed questions for a few rare classes, 894 TREC 8 and TREC 9 questions, and also 500 questions from TREC 10 which serves as our test set3. These questions were manually labeled according to our question hierarchy. Although we allow multiple labels for one question in our classifiers, in our labeling, for simplicity, we assigned exactly one label to each question. Our annotators were requested to choose the most suitable class according to their own understanding. This methodology might cause slight problems in training, when the labels are ambiguous, since some questions are not treated as positive examples for possible classes as they should be. In training, we divide the 5,500 questions from the first three sources randomly into 5 training sets of 1,000, 2,000, 3,000, 4,000 and 5,500 questions. All 500 TREC 10 questions are used as the test set.

**Experiments and Results**

We predict for both coarse and fine labels. Even though we use the same feature space, tools used to extract features are different. Models used to evaluate are also different. We believe that’s the reason behind this variation. We could see the performance variance in our model and li-roth. It outperforms in few scenarios too. Our own naïve bayes model also affects the model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | LiRoth – P1 | Our Model – P1 | LiRoth - | Our model |
| Coarse Classes | 91% | 87% | 98.90% |  |
| Fine Classes | 84.2% | 90.2% | 95% |  |

**Reference**

1. Question Classifier by Li and Roth <http://www.aclweb.org/anthology/C02-1150>
2. A. R. Golding and D. Roth. 1999. A Winnow based approach to context-sensitive spelling correction. Machine Learning, 34(1-3):107–130.
3. S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, and P. Morarescu. 2001. Falcon: Boosting knowledge for answer engines. In Proceedings of the 9th Text Retrieval Conference, NIST
4. S. P. Abney. 1991. Parsing by chunks. In S. P. Abney R. C. Berwick and C. Tenny, editors, Principle-based parsing: Computation and Psycholinguistics, pages 257–278. Kluwer, Dordrecht.
5. A. Carlson, C. Cumby, J. Rosen, and D. Roth. 1999. The SNoW learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department, May.
6. C. Cumby and D. Roth. 2000. Relational representations that facilitate learning. In Proc. of the International Conference on the Principles of Knowledge Representation and Reasoning, pages 425–434.