Hierarchical Question Classification

Alan Joyce
Stanford University
CS 224N: Natural Language Processing

ABSTRACT

Responding accurately to naturally phrased questions has been an ongoing challenge in the fields of information retrieval and natural language processing. In this work, we investigate a task at the core of this question answering process: identifying what type of answer is being requested. By building and evaluating a hierarchical question classifier, we develop an approach to this task that distinguishes among six coarse-grained question classes and fifty fine-grained ones. We measure the effectiveness of various feature types and compare the performance of this hierarchical classifier to that of a traditional, non-hierarchical one, gaining some insight into the viability of such a solution.

1. INTRODUCTION

Earlier this year, IBM's Watson computer system stunned the world by defeating two expert human players on the television quiz show *Jeopardy!*. Watson is perhaps the most widely known example of a computer-based question answering system, but the problem of interpreting and responding to natural language questions has been an area of investigation among researchers for decades.

One of the efforts that has helped to catalyze this investigation is the Text REtrieval Conference (TREC) competition, an exercise in open-domain question answering that challenges participants to retrieve answers to plain English questions by extracting them from a large library of static documents. This is a broad and complex task, but many researchers realized that they could considerably simplify the answer space for any given question if they first classified it by answer tag. In [1], IBM's system uses a collection of 31 such tags to classify a question by the type of answer it requires, an approach that is mirrored in an array of other TREC entries, beginning around 2001.

It is easy to see how this classification approach can benefit the task of question answering. Consider the question "*What is the largest city in the world?*". If a system needs to search through a large document collection to find the answer, this task will be greatly simplified by knowing that the object of the search is a location. Even better, if the system can determine that the question is asking specifically for a city, then the search space will be narrowed significantly.

The above example is doubly informative because it also motivates the idea of a classification hierarchy. Ideally, we would like to know the fine-grained classification of "city", but this classification is also just a specific case of the "location" classification. By first making the broader judgement that the question is asking for a location, we can then use this information to drill down and determine that a city is the specific type of location being asked for.

2. RELATED WORK

This work is based directly upon the efforts of Xin Li and Dan Roth [2] in their design of a hierarchical question classifier for the TREC 10 question answering competition. Li and Roth use a hierarchy consisting of fifty fine-grained classifications and six coarse-grained ones, along with a machine learning architecture with local and semantic features. They are able to eventually produce a system that can assign both coarse and fine-grained classifications with greater than 90% accuracy.

Additionally, Li and Roth investigate the relative benefits of local and semantic features in their classification, while also assessing the performance of a flat classifier in contrast to their hierarchical one. They find that semantic information provides a noticeable benefit over simple local features, but they are unable to identify any significant advantage that their hierarchical classifier possesses over a flat, non-hierarchical one. In contrast to their expectations, Li and Roth find that the added simplicity of picking a fine-grained classifier based on course-grain information is counteracted by an increased degree of inaccuracy on the coarse-grain level.

Of course, the work by Li and Roth does not stand alone as an approach to the type of question answering task presented in TREC-10. Rennert [3] proposes a set of question categories, such as "HOW MANY" and "WHAT IS NAMED", which represent a similar classification methodology, but with a question-centric approach rather than an answer-centric one. Additionally, Chen et al. [4] utilize question types such as "What/Which" and "Name-of", while Monz and de Rijke list similar question types, including "thing-def" and "also-known-as" [5].

IBM's system [1] defines a hierarchy similar to Li and Roth's, but one that is less extensive and not as strictly followed. Soubbotin's approach [6] is focused on identifying indicative patterns, but still makes room for a component that identifies the type of question being asked (e.g. "When (What year) born"). Taking a more direct approach, Ferret et al. [7] classify most questions with a list of named entities that could fit as an answer (e.g. "PERSON", "ORGANIZATION"). Finally, Plamondon and Lapalme [8] use extraction functions such as "measure(a, b)" and "location(a)" to identify the various types of question.

All of these methodologies demonstrate a healthy variety of potential approaches for question classification, and often for question answering as well. However, work on the problem of question classification has also extended beyond what was seen in TREC-10. Zhang and Lee [9] present an evaluation of five machine learning algorithms, and find that support vector machines, combined with the same question hierarchy defined by Li and Roth, produce the best classifier among the methods tested.

Blunsom et al. [10] utilize maximum entropy models, along with the same dataset and hierarchy as Li and Roth, to show that a hierarchical question model combined with a maximum entropy classifier produces a new state of the art for question classification. The authors were able to show noticeable improvements over the results obtained by Li and Roth, lending significant appeal to the introduction of maximum entropy models.

Finally, Huang et al. [11] provide some wider context on question classification by evaluating its usefulness as a stepping stone on the path to question answering. Unsurprisingly, question classification is determined to be an essential part of a competent question answering system, reaffirming the overall importance of the task.

3. APPROACH

In this work, we do not attempt to pursue a new state of the art for question classification. Instead, we endeavor to use existing methods and tools to produce a competent hierarchical question classification system which we can use to draw insights about such systems.

The datasets used for training and testing the classifier are the same as those used by Li and

Roth for the TREC-10 competition, allowing us to use the 500 actual TREC-10 questions as the test set. Additionally, the question classification hierarchy are the same as that used by Li and Roth, structured as follows:

Abbreviation: abbreviation, explanation
Entity: animal, body, color, creative, currency, disease, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word
Description: definition, description, manner, reason
Human: group, individual, title, description
Location: city, country, mountain, other, state
Numeric: code, count, date, distance, money, order, other, period, percent, speed, temp, size, weight

In accordance with the findings of Blunsom et al., the machine learning algorithm used to build the question classifier is a maximum entropy model. The basic implementation is provided by the Stanford Classifier [12], which is then extended to support the multiple steps of hierarchical classification.

As was indicated in previous work, it is important to use a combination of local and semantic features when building a hierarchical question classifier. In order to satisfy this criteria, we include the use of a part-of-speech tagger [13] and named entity recognizer [14], both of which supply additional features for the maximum entropy classifier. The result is that we have three feature columns available to us throughout the classification process: the literal text of the question, the part-of-speech tag sequence associated with the question, and the list of named entity types seen within the question. Additionally, when building the fine-grained classifier, we have the previously assigned coarse class available to us as a fourth feature column.

4. MEASUREMENTS

There are two stages to building the hierarchical question classifier: coarse classification and fine classification. We focus first on coarse classification as an independent task, building a classifier that can reliably assign one of the six broad classes to a question. We do this first because the performance of fine classification is heavily dependent upon the performance of coarse classification, as each fine class is associated with only one coarse class. Thus, assigned coarse classes become important features in the process of selecting a fine classification. Once we have produced a competent coarse classifier, we can begin to build the fine classifier, aiming for a system that can reliably assign one of the fifty specific classes to a question. Both stages of this process are exercises in feature manipulation and evaluation, tweaking the methods for selecting features and observing the effects on classifier performance.

Coarse Classification

We begin with a simple model for feature selection on our three columns of input (question text, part-of-speech tags, and named entities). All columns are parsed for n-gram features, with specific features for prefix and suffix strings. This results in good average performance across all classes ($F1_{macro}$ = 0.82), but the classifier struggles with the Entity class ($F1_{ENTY}$ = 0.69).

In order to improve performance, we revisit a simplification made in the initial model: the identical treatment of all input columns when extracting features. In reality, the first two input columns are direct representations of the sentence that maintain its structure while the third input column is simply a list of entities. To account for this, we introduce a new model, which treats each word of the named entity column as a single feature and disregards the ordering of the words. Performance here shows an improvement overall ($F1_{macro}$ = 0.83), and an improvement in the Entity class ($F1_{ENTY}$ = 0.71).

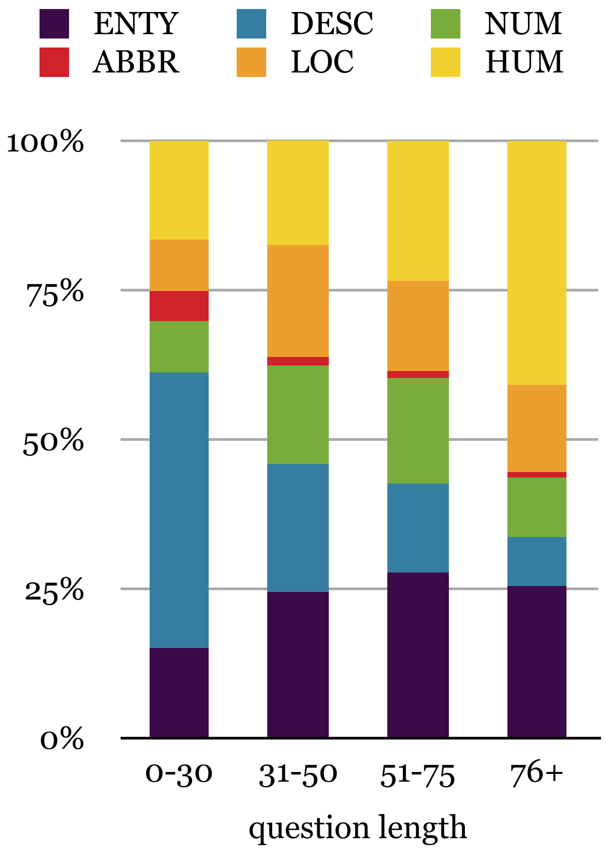

Figure 1: Question class distribution across four length buckets.

As an experiment, we also investigate whether question length is a valuable feature by binning the question text into one of four length buckets: 0-30, 31-50, 51-75, and 76+. A few potentially important trends do result from this process (see Figure 1), most notably that questions of length 30 or less are significantly more likely to be of class Description than any other class and questions of length 76 or more are more likely to be of class Human. However, these insights appear on the fringes of the data, leaving overall performance generally unchanged by this addition.

Before moving on to the process of fine classification, it will be instructive to step back into a more general view and assess the importance of the individual feature columns themselves toward building a competent classifier. We break our process down into three separate models: one based upon only question text (M1), one based upon question text and part-of-speech tags (M2), and one based upon all three columns (M3). We evaluate the performance of each of these models in order to view the contributions of each feature column.

	M1	M2	M3
$F1_{micro}$	0.81	0.82	0.82
$F1_{macro}$	0.82	0.83	0.83

Table 1: Performance metrics for models M1, M2, and M3 for a coarse classifier.

While part-of-speech tagging appears to provide an improvement, the value of identifying named entities is called into question. With that said, we will revisit these contribution metrics in the fine classification step before drawing any large-scale conclusions about feature column contributions.

Fine Classification

We now enter the second stage of the hierarchical classification process by taking our coarse tag results from the first step and using them as input for a second round of classification. We still use the same dataset to train this classifier, but the gold answer is now a fine-grained class (e.g. “city”, “plant”, “count”) and the correct coarse-grained class is used as a feature column in the training process.

We use the same feature extraction model as we did for coarse classification, but with the addition of a zero column that contains the corresponding coarse class for each question. We interpret this column as a simple string, creating a set of six features that correspond to the collection of coarse classes.

The resulting classification performance is moderate, with an average of $F1_{micro}$ = 0.70 across all test questions. At the same time, the average across all question classes is surprisingly low at $F1_{macro} = 0.35$. This is not a drastic cause for concern, however, as it is primarily a consequence of the TREC-10 test set not containing examples for every fine class, and only containing a small number of examples for a few classes, yielding a high number of classes with F1 = 0.00.

	P	R	F1
abbreviation	1.00	1.00	1.00
language	1.00	1.00	1.00
technique	1.00	1.00	1.00
country	1.00	1.00	1.00
date	0.90	0.92	0.91
count	0.81	1.00	0.90
color	1.00	0.80	0.89
individual	0.78	0.96	0.86
definition	0.74	1.00	0.85
reason	0.83	0.83	0.83
explanation	0.83	0.63	0.71
other	0.57	0.80	0.66
city	0.89	0.44	0.59
state	0.75	0.43	0.55
substance	1.00	0.33	0.50
speed	1.00	0.33	0.50
mountain	0.50	0.33	0.40
animal	0.42	0.31	0.36
food	0.50	0.25	0.33
distance	1.00	0.19	0.32
description	0.30	0.30	0.30
plant	0.50	0.20	0.29
manner	0.17	1.00	0.29
money	0.25	0.33	0.29
group	0.33	0.17	0.22

Table 2: Performance for fine-grained classifications. Classes with F1 = 0.00 are not shown.

As shown in Table 2, performance across individual classes ranges widely, and with no definitive trend. Certainly, some classes with easily identifiable question formats (date, count, individual) rise to the top, but other classes which would seem similarly easy to identify (distance, money) are among the lowest performers. We will investigate these discrepancies further in our error analysis.

For now, we now revisit the evaluation of feature columns by replicating the models M1 (question text only), M2 (question text and part-of-speech tags), and M3 (all columns) within the fine-grained classifier.

	M1	M2	M3
$F1_{micro}$	0.67	0.70	0.70
$F1_{macro}$	0.34	0.35	0.35

Table 3: Performance metrics for models M1, M2, and M3 for a fine classifier.

As seen with the coarse classifier, the addition of named entity tags does not appear to have a significant effect on performance metrics. However, the value of part-of-speech tagging is further confirmed by this case.

Comparison with a Flat Classifier

In order to gain additional insight into the effectiveness of our hierarchical classifier, it is a valuable exercise to compare its performance with that of a flat (non-hierarchical) classifier. To test this, we create such a flat classifier by modifying our coarse classifier to use fine class tags as gold answers instead of coarse class tags.

Running this flat model over the dataset results in $F1_{micro} = 0.69$ and $F1_{macro} = 0.35$. This result is somewhat surprising, but ultimately in line with the findings of Li and Roth in that the hierarchical classifier does not display a significant performance gain over a flat classifier.

5. ANALYSIS

With a maximal F1 of 0.83 for coarse classification and 0.70 for fine classification, a degree of competence is established, but it is clear that there is still some ground to be covered toward refining and improving the hierarchical classifier detailed here. Thus, we present an analysis of the successes and failures of the current system over both stages of classification.

Coarse Classification

In observing the process of coarse classification, one class stands out above the rest as especially easy to classify: Human. Perhaps it is natural that human language uses such a unique encoding for referencing humanity, but the use of heavily-weighted features based on the word "Who" (and a few moderately-weighted features based on the presence of a past-tense verb) allows classification of Human questions with F1 = 0.90.

Other high-performing classes include Abbreviation (F1 = 0.88), Number (F1 = 0.85), and Description (F1 = 0.84). Among these, Number is associated with the word "When", while Description is paired off with "How" and "Why". Abbreviation does not utilize any obviously significant features, but it is a relatively rare class (only 9 positive examples in the test set).

On the other end of the spectrum, Entity performs the most poorly (F1 = 0.71) and Location ranks second-to-last (F1 = 0.81). The somewhat low performance of Location is surprising, as it is associated with the rather precise word "Where", the terms "city" and "county", and the named entity "LOCATION". Location misses a high number of false negatives (21) in the dataset, often when the word "What" is used. Questions such as "*What is the capital of Yugoslavia?*" and "*What continent is Egypt on?*" are interpreted as Description or Entity, while the question "*What is the length of the coastline of the state of Alaska?*" is falsely identified as a Location when it should be a Number. These phrasings can be rather ambiguous, so it is easy to see why the Location classification sometimes misses the mark.

As for Entity, the issue appears to be a case of excessive broadness. Fine classes under Entity range from "currency" to "sport", making it difficult to find reliable signals in the question texts. Indeed, the classifier identifies hardly any high-weight features for Entity, and what features do exist are not very specific (e.g. "sh", "olo", and "ar of"). Look at an ambiguous question like "*What is another name for vitamin B1?*", which is mistakenly classified as a Description instead of an Entity, and it is apparent that correctly applying the Entity tag is a difficult task.

Overall, the majority of coarse classes are identified reasonably well through a selection of well-identified features. Of course, the words "Who", "What", "When", "Where", "How", and "Why" play the biggest role in differentiating question classes, but classes such as Location and Human manage to use other high-weight features to their advantage as well.

Fine Classification

Within the realm of fine classification, high-weight features become significantly more specific and overall performance sees a noticeable drop, but still remains competent. At the same time, a high number of fine classes (20) have F1 = 0.00 within the TREC-10 dataset.

In keeping with the results of the coarse classification stage, "individual" sees relatively successful results (see Table 2). This is again due to the obvious feature "Who", and the presence of a past-tense verb. However, "individual" is outperformed by "date" and "country", which exploit the words "When" and "country" respectively to develop effective features. This trend continues with "city" and "state", both of which leverage their names as basic but mostly effective features.

On the less successful end, we have the “manner” class, which exhibits some interesting behavior that nets it a perfect recall score, but precision of only 0.17. This is because “manner” manages to claim the feature “How” with a shockingly high weight, as well as the Wh-adverb part-of-speech tag. This overconfidence results in a 5:1 ratio of false positives to true positives, with questions like “*How tall is the Sears Building?*” and “*How old was Elvis Presley when he died?*” being mistakenly classified as “manner”.

Other low-performing classes include the surprising “food” and “plant”, which exhibit surprisingly low recall (0.25 and 0.20, respectively). It would be natural to assume that specialized subject areas such as these would have specific and high-weight words that would distinguish them. Indeed, “food” is associated with the feature “eat”, which does bring it some benefit. However, it appears that overall the classifier was simply not exposed to enough examples of the jargon associated with these classes to properly identify them. A question like “*What fruit is Melba sauce made from?*” is mistakenly identified as “plant” when the word “sauce” should strongly indicate “food” instead.

Finally, the other unfortunate trend among the results was a high number of fine classes with F1 = 0.0. While most of these were the result of the TREC-10 dataset containing few or no examples of the class, the tags “vehicle”, “temperature”, “period”, and “term” all had a significant number of false negatives (more than 5 each).

Again, several of these appear to be the result of the classifier not being familiar with the necessary associated terms. The question “*What were Christopher Columbus’ three ships?*”, along with a few other “ship”, and “plane” questions should fall into the “vehicle” class, but are designated as “other” instead. At the same time, somehow the word “temperature” is not associated with the class “temperature”, with several questions explicitly using the word and being classified as “other” or “definition”. This is the sort of feature that would lend itself well to either manual addition or the acquisition of further semantic knowledge in the relevant subject area.

Other classes that could benefit from some manual feature addition include “term” and “period”. The question “*What do you call a newborn kangaroo?*” is classified as “animal” instead of “term”, but this and many other questions could have been identified properly if the phrases “What do... call” and “What is... name” were features. Similarly, the question “*For how long is an elephant pregnant?*” is assigned the “count” tag when the phrase “For how long” should suggest “period”.

Despite all of these less successful classes, the fine-grained classifier did a reasonable job of finding good, specific features for a number of classes. The overarching message here was a need for additional semantic context within the linguistic domains of individual fine classes, as well as a potential need for a handful of well-crafted manual feature additions or modifications.

6. CONCLUSIONS

While the hierarchical question classifier presented here certainly does not improve upon the state of the art, it does manage to be basically competent and there are some useful insights to be drawn from its development and evaluation.

Based on the evaluation of various feature columns and extraction mechanisms, we can assert that question text and part-of-speech tags appear to be valuable sources of classification information, while the inclusion of named entities could not be shown to have a significant effect on performance. Additionally, question length has the potential to be a useful feature in a few special cases, but overall does not appear to produce a noticeable benefit either.

In comparing this hierarchical classifier to a flat, non-hierarchical one, we observe that the two models achieve roughly equivalent performance (within 1%). Thus, it is unclear whether the additional effort and resources required to implement a hierarchical model are a wholly worthwhile investment.

There are a variety of refinements and extensions that could be incorporated into the system detailed here. The most notable of these is likely the inclusion of additional semantic information in the feature extraction process — a classification such as "food" or "plant" would be greatly aided by knowledge of semantically related word sets, as shown by Li and Roth. On another front, it would be interesting to replace the manually defined hierarchy of question classes with a supervised learning model, whereby classes could be extracted from any corpus of questions. This approach could allow for extensions into more specific, jargon-heavy question domains, in addition to benefiting the general usefulness of a hierarchical question classifier.

7. REFERENCES

[1] Ittycheriah, Abraham, et al. IBM's Statistical Question Answering System — TREC-10, 2001.

[2] Li, Xin and Roth, Dan. Learning Question Classifiers, 2002.

[3] Rennert, Philip. Word proximity QA system, 2001.

[4] Chen, Jiangping, et al. Question Answering: CNLP at the TREC-10 Question Answering Track, 2001.

[5] Monz, Christof and de Rijke, Maarten. Tequestra: The University of Amsterdam's Textual Question Answering System, 2001.

[6] Soubbotin. M.M. Patterns of Potential Answer Expressions as Clues to the Right Answers, 2001.

[7] Ferret, O., et al. Finding an answer based on the recognition of the question focus, 2001.

[8] Plamondon, Luc and Lapalme, Guy. The QUANTUM Question Answering System, 2001.

[9] Zhang, Dell and Lee, Wee Sun. Question classification using support vector mechanisms, 2003.

[10] Blunsom, Phil, et al. Question classification with log-linear models, 2006.

[11] Huang, Zhiheng, et al. Investigation of question classifier in question answering, 2009.

[12] Manning, Christopher and Klein, Dan. Optimization, Maxent Models, and Conditional Estimation without Magic, 2003.

[13] Toutanova, Kristina, et al. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network, 2003.

[14] Finkel, Jenny Rose, et al. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling, 2005.