Question Classification using Latent Focus Words

Anand Iyer
Stanford University
Computer Science
iyera@stanford.edu

Ritvik Mudur
Stanford University
Computer Science
rmudur@stanford.edu

ABSTRACT

In this paper we present a novel approach to the problem of selecting question focus words (QFWs) for the task of question classification. We show that by starting with a random choice of focus words, we can train a system that is comparable to one based solely on heuristics. The algorithms explored provide a more elegant approach to the problem, and one that is likely to scale better to larger data and finer question categories. The results show promise for exploring more avenues with such an approach, and systems that can integrate latent QFWs with strong heuristics. .

1. INTRODUCTION

Question classification refers to the task of assigning a given question to one (or more) of a set of pre-defined question criterions. Question classifiers play a significant role in the open-domain question answering (QA) task. With free-form questions and a large knowledge-base of text data it becomes crucial to constrain the space in which to search for an answer. Cheaper and faster storage and increasing amounts of data will likely make the contribution of question classifiers even more significant. A pre-set question type assigned to an input question can allow for more efficient passage-retrieval (PR) and final answer extraction (AE).

Classifiers are trained in the question processing step (QP) of a QA system. Incoming questions are parsed and linguistic features are extracted that are used to train the question classifiers. Li and Roth [1] developed a system with 6 coarse and 50 fine question classes, and used a mixture of syntactic and semantic features to train a hierarchical classifier. In later work [2], explicitly identifying the word (or phrase) indicating the main focus of the question was shown to be an advantageous approach in statistical question classification. The extraction of these question focus words (QFWs) has mostly used heuristics. For example in [2], the focus word is determined to be the head word (based on POS tags) of the main noun phrase in the question. Another approach is to learn the focus words using a statistical model, and essentially learn weights for potential focus words (or positions) as in [3][4]. However, such an approach requires knowing the focus words as part of the training set, or being able to estimate it with high certainty.

Another approach is to consider the QFWs as hidden words in our question training set. These latent variables can then be learned by solving an optimization problem that essentially results in the strongest classifier (i.e. one with the most confident and accurate labels). The main advantage this approach has over the other two is scalability. Applying QA systems to a larger stream of topics will require additional (finer) question classes. As the number of classes begins to grow, it becomes impractical to develop specific heuristics for various subsets of question categories. Moreover, labeling the focus words in datasets can be a time-consuming (and hence expensive) task, considerably more so than just general class-labels. With larger databanks of questions, it becomes unreasonable to expect marked focus words in our training sets. Neither of these issues is a concern for latent variables approach, thus allowing it to scale nicely with more classes and additional data.

In the remainder of this paper, we outline our approach to learning latent focus words for the task of question classification. We explain our training and testing methodologies, the experimental setup and results, and discuss the promise that latent learning holds in a QA system.

2. LEARNING LATENT FOCUS WORDS

The aim of any latent learning algorithm is to find the goal (optimal) feature vector for a given training sample. The optimal feature vector is one that results (when trained) in an accurate labeling by the classifier, and does so with the largest possible confidence (or score). As such for a complete training set, we wish to find the best set of feature vectors that return accurate labels with the maximum possible score. This should lead us to train a stronger classifier overall.

2.1 Training with Latent Focus Words

We adapt the latent training algorithms explained in [5][6] to the question classification task, These algorithms use a two step iterative process (similar to Expectation Minimization) that attempts to minimize the difference between the score (or margin) of the goal feature vector and that of the current feature vector. We detail our training procedure below:

Table 1 – Two step training algorithm to find latent focus words for a multiclass-classifier C

Training procedure
➢ Initialize by selecting a set of random QFWs for each training sample and training classifier C ➢ Loop until a stopping condition is met: 1. Estimate the best QFW in the current iteration as: $$qfw^i = \underset{qfw \in QFW(x_i)}{\operatorname{argmax}} F(y_i * \mid x_i, qfw)$$ 2. Use the focus words determined in 1, to generate new features for our training samples and re-train our classifier C

In our first step, we loop through all words in a given training question that fall within our space of latent QFWs (discussed in section 2.3), generate appropriate features for each and find the one that returns the maximum score (F) on the gold label (y_i^*) with our multi-class classifier C. For the second step, we simply use the feature vectors with QFWs selected in the previous step to re-train C.

The calculation of function *F* would differ based on the form of C. For example, if we were using SVMs for classification *F* would represent the distance to the separating hyperplane.

2.2 Testing with Latent Focus Words

It isn't clear how one should approach the selection of QFWs for test instances with a system trained on latent focus words. We would like the system to select the appropriate focus word for feature generation such that: a) it is consistent with our training, i.e., features selected for training samples in the testing procedure should be the same as in the final iteration of training, and b) the focus word selected is the most likely one given our trained model. One approach is to train an additional model to guess the most likely focus word for a given sentence based on the focus words found during training. For example, we could use maximum likelihood estimation to determine a distribution over a subset of our features for

each latent focus word (by simply counting the number of times a focus word is selected for a given set of features). However, we take a rather simpler approach to the problem and simply use the word that returns the largest score across all labels in our classifier. The procedure is as follows:

Table 2 - Testing procedure with latent focus words

Testing procedure
➢ Return the best label y_i given by: $y_i = \underset{y \in Y}{\operatorname{argmax}} \left(\underset{qfw \in QFW(x_i)}{\operatorname{argmax}} F(y \mid x_i, qfw) \right)$

Y represents the set of all possible labels, and so the procedure above returns the most confident guess for a given question by looping across all focus words and returning the value of the label classified with the maximum score.

2.3 Space of Latent Variables

There are many words in a question that do not add sufficient value to the goal of classification. However, if we do not limit the space of our latent focus words these might get picked as QFWs. Examples are words such as 'as','the','are','a','is',etc. These are what are known as "Stop-Words" in Informational Retrieval terminology. We do not consider these "Stop-Words" as potential QFW candidates.

We also do not consider question words such as "What","How","Where","When","Why","Which" (i.e. interrogative words) as QFWs. These words are the strongest indicators of the questions' coarse category. Hence these words, and features on these words, are treated separately. The QFWs are supposed to provide information in addition to that provided by the interrogative word, and essentially help the classifier determine the finer sub-categories a question may fall in. Hence it would be redundant if an interrogative word was also chosen as a QFW.

Commonly used heuristics ([2]) would suggest using the head words in noun and verb phrases (NP, VP) as focus words; however, this was too restricting since there are a non-trivial number of examples where the QFWs are adjectives, adverbs,etc. In fact, the system we compare our results to and build on uses a set of templates in addition to the simple NP/VP heuristics to extract its focus words (see 3.2. for details).

2.4 Using Ensemble Learning

There are many question types where we only have a handful of training examples. When we initialize our system and pick QFWs randomly, it is possible that we do not pick the correct QFW at all (or not sufficiently enough). Then for the next iteration, it is unlikely that the trained classifier will pick the correct QFW since it will have a weak score. After all, the classifier never saw a positive example where the features corresponding to that word as QFW were turned on (or did not see sufficient positive examples where those features were turned on).

For example, consider the question class **ABBR:abb**. Good QFWs for questions of this class are the words “abbreviation”, “abbreviated”, “acronym”, etc. In the training examples (see 3.1 for details), acronym occurs just twice. It is very likely that, in the first iteration the word “acronym” is not picked for the two examples that contain it. As such the classifier never encounters a positive example of

ABBR:abb with “acronym” as the QFW in its first iteration. Hence, in the next iteration it is unlikely that the classifier will pick the word acronym as the QFW which gives highest confidence for the label.

In order to remedy this problem we use an ensemble of N classifiers in our first iteration, where the QFWs for each classifier and each question therein are chosen at random. Thus different classifiers will likely be trained with different QFWs for the same question. In step 2 of the first iteration (Table 1) we use a voting scheme to determine the focus word for a question. The word with the most votes for a given question is selected as the QFW.

2.5 Multiple QFWs

We also explore the idea of using multiple QFWs for a given sentence. There are often cases when the choice of QFW is ambiguous, and multiple words seem likely to be potential focus words. For example, the question:

```
Which university's football team won the 2009 Superbowl?
```

This question is classified as HUM:gr i.e. it is looking for information related to an organization or another similar collective group of individuals (such as a team, a company, a band,etc). There are two words in this question that are useful as QFWs since both these words are indicative of an organisation: univeristy and team. Thus it seems useful to add two QFWs, rather than just adding either one alone. QFWs often also impact the answer extraction procedure in some QA systems where text containing the focus word is given emphasis ([7]) and additional words can prove beneficial.

In our system, we define ambiguity as the situation where two words produce classification scores that are extremely similar (i.e. within a pre-defined threshold).

3. EXPERIMENTAL SETUP AND RESULTS

3.1 Data and Code Base

Our implementation is based on the question classifier written as part of the QA package produced by Surdeanu et al. for [7]. The training data used includes labeled questions from the TREC – QA track, and the test set comprises of TREC 10 questions ([1]).

3.2 Experiments Run and Features Used

We ran several experiments, using different combinations of features for training. In order to judge the raw performance of our algorithm, we strip down all features except those relating to QFWs (see [7] for details). We also test the system using all the features specified in [7]. Both a multi-class (one v/s rest) perceptron and SVM were used as classifiers.

As a stopping criterion we attempted to count the total number of changes to the focus word selections in each iteration and set a threshold. However, the algorithm appears to converge quickly and in the interest of time we only ran each experiment for a fixed number of iterations (n=10).

The results of our system are compared against those of [7], which uses heuristics for QFW selection. In [7], the QFWs are picked based on templates of several styles of questions. For example the question:

```
“What relative of the racoon is sometimes known as the cat-bear?”
```

falls under the template “`What|Which type|kind|relative of`”, which is used to set the position of the focus word to be the head word of the 3rd chunked token. In this case it happens to be “the raccoon”. Similar templates are used for a variety of question styles, and when a question is not captured by the above templates the head word of the first NP or VP (whichever comes first in the question) is used as the focus word.

3.3 Results

We present a summary of our results on the test set when using only QFW features and all features described in the question processing section of [7].

Table 3 – Test set accuracies using only QFW related features (%)

	Perceptron	**SVM**
Original Heuristics	81	79.8
Latent FWs w/o Ensemble	71.2	72.6
Latent FWs w/ Ensemble	73.4	-
Mutliple QFWs	69.3	-

We limited our tests on the full feature set to only the perceptron, and did not test the multiple QFWs approach due to poorer performance. Similarly SVMs took considerably longer to run but still only produced comparable results, and were thus not included in the table below.

Table 4 - Test set accuracies using all features in [7] (%)

	Perceptron
Original Heuristics	87.6
Latent FWs w/o Ensemble	82.6
Latent FWs w/ Ensemble	85.8

4. DISCUSSION

As summarized in the Tables 3 and 4 above, our best performance still falls short of the original heuristics. However, when examining the results we find numerous cases where the latent learning selects more significant focus words. Some examples of these are shown in Table 5 below.

Table 5 - Sample QFW results from our ensemble trained system

Question	**Type**	**QFW w/ Original Heuristics**	**QFW w/ Latent learning (w/ Ensemble)**
Who was the first person to reach the North Pole ?	HUM:ind	reach	person
Where is John Wayne airport ?	LOC:other	is	airport
What is the name given to the Tiger at Louisiana State University ?	ENTY:animal	given	name
What is the heaviest naturally occurring element ?	ENTY:substance	occurring	element
Who is the only president to serve 2 non-consecutive terms ?	HUM:ind	serve	president

Where our system fails often is when pitted against template based matches. Superior focus words are often selected using heuristics when the question style matches a particular template. The templates (as discussed in 3.2) are formulated such that the focus word can be picked with extremely high certainty (given that the question itself has a correct grammatical structure). For example, the question

```
What type of polymer is used for bulletproof vests ?
```

results in the template "`What|Which type|kind|relative of`", selecting the QFW "polymer" while our system selects "used". In fact, our experiments showed us that nearly half of all test set questions fell into one of these very templates.

However when using all features of the system proposed in [7], our system produces reasonable results. They are slightly poorer but comparable, since the larger feature vector likely compensates for a few sub-optimal focus word choices. What a latent learning approach allows us to exploit the most though, is actually ambiguity. There maybe several cases when labeling focus words where a choice must be made. In such ambiguous cases, it might be advantageous for classification purposes (or a complete QA system) to pick one word over the other. However, knowing which word will help classification beforehand is really not possible. It seems more appropriate to delegate this to a learning algorithm which selects the most appropriate focus word to maximize classification performance, and this is exactly what a latent learning approach affords us. We believed multiple QFWs would enhance performance in such ambiguous situations; however it's weaker performance is puzzling and should be analyzed further.

5. FUTURE WORK

We would like to stress that the work presented here should be considered similar to a first run experiment. There is scope for improvement, and the right mix of features and techniques will likely boost the performance of a classification system that uses latent focus words.

Given the increase in performance seen by using a simple ensemble in our very first iteration, it might be worth exploring more sophisticated ensemble learning algorithm such as Boosting. Since our system fails to select appropriate words on some simple templates, it might be worthwhile exploring the addition of higher order N-grams in the focus word features. Our current implementation was limited to tri-grams; however intuitively, a 4-gram should be able to better capture templates of the type "What type of <QFW>" or "Can you name <QFW>".

Another potential approach is combining Templates (as described in 3.2) and the latent variable approach. We can assign a confidence level to templates. Certain templates are bound to perform better than others. Based in the performance of templates (or a held out validation set), we can assign confidence scores to these templates. When making the final choice of QFW for a test question, we can aggregate the following to make a better choice:1) the word picked by the template and the confidence score of the template, (2) the confidence scores returned by our Latent variable method, for the different QFW choices.

6. CONCLUSION

We have shown that the problem of selecting question focus words can be treated using latent variables. By starting with a random initialization of focus words for each of our training instances we were able to build a system with reasonable performance when pitted against another based solely on heuristics. Despite a slightly weaker performance, latent variables still present a more elegant approach to focus word selection and one that can exploit the ambiguity in focus word labeling while also making the question classification system easier to scale.

7. ACKNOWLEDGMENTS

We would like to thank Mihai Surdeanu for all his guidance and support. We are grateful for his cooperation and providing us with his package system code.

8. REFERENCES

[1] Xin Li, Dan Roth, Learning Question Classifiers. *COLING'02*, Aug., 2002..

[2] Metzler, D. and Croft W. B. (2005). Analysis of Statistical Question Classification for Fact-based Questions. *Information Retrieval*, vol. 8, 481-504.

[3] Z Zhang, Y Zhang, T Liu, S L. Automatic Recognition of Focus and Interrogative Word in Chinese Question for Classification. Computer and Information Science, Vol 3, No 1 (2010)

[4] Ramakrishnan, G., Paranjpe, D., and Dom, B. 2005. A structure-sensitive framework for text categorization. In *Proceedings of the 14th ACM international Conference on information and Knowledge Management* (Bremen, Germany, October 31 - November 05, 2005). CIKM '05. ACM, New York, NY, 337-338. DOI= http://doi.acm.org/10.1145/1099554.1099655

[5] Yudong Liu, Gholamreza Haffari, Anoop Sarkar. Latent SVMs for Semantic Role Labeling using LTAG Derivation Trees. In Proceedings of Human Language Technologies: the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010), Los Angeles, June 1-6, 2010.

[6] Sun, X., T. Matsuzaki, D. Okanohara, and J. Tsujii (2009). Latent variable perceptron algorithm for structured classification. In Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009).

[7] Mihai Surdeanu, David Dominguez-Sal, and Pere Comas. *Performance Analysis of a Factoid Question Answering System for Spontaneous Speech Transcriptions*. Proceedings of the 10th International Conference on Speech Communication and Technology (Interspeech), September 2006