Active Learning for Building a Corpus of Questions for Parsing

Jordi Atserias[1], Giuseppe Attardi[2], Maria Simi[2], Hugo Zaragoza[1]

[1] Yahoo Research, Barcelona, Spain

[2] Dipartimento di Informatica, Università di Pisa, Italy

Abstract

This paper describes how we built a dependency Treebank for questions. The questions for the Treebank were drawn from questions from the TREC 10 QA task and from Yahoo! Answers. Among the uses for the corpus is to train a dependency parser achieving good accuracy on parsing questions without hurting its overall accuracy. We also explore active learning techniques to determine the suitable size for a corpus of questions in order to achieve adequate accuracy while minimizing the annotation efforts.

Introduction

Treebanks for training dependency parsers are becoming popular through the activities of the CoNLL Shared tasks [CoNLL]. However most of these corpora are obtained from newspaper articles or Web pages. These sources rarely contain question sentences and therefore the parsers trained on these corpora have poor accuracy in analyzing questions. Question analysis is required though in many applications, for instance Question Answering or Text Entailment.

The TREC Question Answering task involves analyzing questions and many systems perform a grammatical analysis of questions, but annotated questions are not generally available. Hermjakob (2001) created his own resource annotating the questions in Penn Treebank style with constituent parse trees. The Childes Corpus (CHILDES) contains questions annotated with dependencies, but the questions are of a type hardly comparable to the ones that a user would post to a Question Answering system.

Yahoo! Answers is a popular service, which provides a meeting place where users can look for advice from experts, who can be just other users. Yahoo! has collected several million of questions in many languages and makes part of this collection freely available on request for research purposes through the Yahoo! Webscope program. Linguistic analysis of these sentences would be quite useful for building applications that exploit the rich knowledge that questions and associated answers provide.

A naïve attempt at parsing questions with a parser trained on the Penn Treebank achieves an accuracy of approximately 86% (measured on our question test), after training on the complete Penn Treebank training corpus. This result is quite disappointing comparing to the 89% performance that can be achieved on classic benchmarks for the Peen Treebank. One reason for the lower accuracy is that the Penn Treebank contains few questions (we counted just 3,553 questions, about 0.75% of the whole sentences), sometimes not even consistently annotated. For instance, while “how” is usually connected to the main verb in expressions such as “how do <subj> <main-verb> …”, producing a non-projective dependency, in the sentence “… how did a senator like this end up approving …” the token “how” is connected to “did”. Moreover the annotation of similar expressions such as “ how much“, “how many”, “how soon” … is not coherent in the corpus: usually, but not always, “much”,

"many", "soon" are annotated as dependents of "how" (and labelled AMOD); for a different style of annotation see for example the question "How much are these benefits worth?".

Thus in order to improve the parser accuracy, a suitable corpus of questions, annotated with dependency relations, is needed. In this work we address these questions: how big a corpus of questions should be in order to achieve adequate accuracy? Is a single corpus adequate to analyze both questions and non-questions?

Question Corpus Construction

We collected unannotated questions[1] from the TREC QA main task (TREC QA) where systems are required to answer 500 short, fact-based questions, as improving the parsing of this kind of factual question could have a direct impact on the current NLP applications.

In order to cover a wider spectrum of general questions we also selected a random sample of about 800 sentences from the Yahoo! Answers Collection [yanswers] (which includes 4,483,032 questions and their corresponding answers).

As in many web corpora, often in Yahoo! answers the questions posted by the users are not grammatically correct or contain forms of expressions like abbreviation, slang or emoticons. We decided to automatically filter the sentences with spelling mistakes or those whose parse trees had multiple roots. The resulting set was first parsed using desr [Attardi 2006] trained on the WSJ; then we corrected the PoS, parsed again with the correct Part of Speech and finally we manually revised the resulting dependency labels.

	Sentences	Avg. Length	Tokens
Yahoo! Answers Corpus	852	10.65	9,080
Trec Corpus	500	7.51	3,758
Question Corpus	1352	9.50	12,839

Table 1: Corpus statistics

Table 1 reports some statistics of the question corpus built.

Questions in English present verb-noun order inversion and non-projective dependencies are quite common, in part as a consequence of this inversion. Differences between the Penn Treebank and the question corpora in the dependency labels distributions can also explain the specificity of the task.

Approach

We addressed the questions stated in the introduction by means of *active learning*. Active learning is an iterative process where a learner is trained using an initial training set and then it chooses some examples from an unannotated collection, so that it can be annotated and added to the training corpus

[1] http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/

for the next iteration. If the selection criterion is effective, a much smaller number of examples needs to be provided to achieve the same level of accuracy than using normal supervised learning.

In classic AL the optimal size of data to add at each criterion is a single pattern. After labeling every pattern we re-compute interestedness of unlabeled points, choose the one with highest, label it, re-train, etc. Adding more than one pattern at a time incurs some loss of information, and as we add more and more in a batch we loose more and more information. In the extreme, if we add all the data at once, we did not do any active learning.

For practical reasons we may want to add more than one pattern at a time, for example if re-training takes a long time and we do not want our judges to wait. In this case, there is a trade-off between how long it takes to re-train and re-compute interestedness, how much can the judges wait, and how much AL power we are willing to "loose". In practice, labeling several points at a time in small batches is a good practice.

Experiments

Our experiments involved using a portion of the Penn Treebank (a random sample of sentences from the CoNLL 2007 English corpus without questions containing 250805 tokens) as initial training corpus (henceforth *base*), and a portion of the Yahoo! Answers as a collection of unlabeled data.

The question corpus was randomly divided in 10409 tokens (1065 questions) for training and 2429 tokens (233 questions) for test. The base corpus was also randomly split between training and test: the training corpus contains 240860 tokens, 9946 sentences; the test set 6494 tokens and 267 sentences.

Random choice

The first criterion we tested is random choice. The experiments show the accuracy obtained by adding, to the *base* training corpus extracted from the Peen Treebank, increasingly bigger subsets of randomly selected questions from the question training set. Table 2 shows the average LAS scores from 5 repetitions (using different seeds) for both the Penn Treebank (base) and Question (quest) test sets. The first column is the baseline using just the Penn Treebank for training, the other columns report the results from adding to the base corpus sentences in 10% increments (about 100 sentences) from the question training set.

	Base	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 9	Step 10	Final
Quest LAS	79.25	85.28	85.54	86.03	86.65	86.75	86.78	87.36	87.61	87.57	86.62	87.61
Base LAS	84.69	85.12	85.46	85.64	85.56	85.39	85.09	86.32	84.61	85.37	84.98	86.60

Table 2: Results for the randomly incremental

The baseline is 79.25%[2] for the question test and 84.69% LAS in the non-question (base) test.

The results show a big boost in accuracy on the question test set with the addition of the first 10%, and then a small increase with the subsequent additions, while the accuracy on the base test set remains almost unaffected.

Likelihood Estimates

We tested more sophisticated criteria to drive active learning based on likelihood estimates of a sentence parse. We used a transition-based parser (Attardi 2006), which uses a classifier to decide which action to perform to carry out parsing. The classifier computes a probability distribution for the possible actions to perform at each step. The likelihood of a parse tree is computed as the product of the probabilities of all the steps used in building the tree. In our experiments we tested three criteria based on sentence likelihood: lowest likelihood of sentence parse tree, highest likelihood and average likelihhod. The questions in the question training corpus were parsed and then ordered a priori with these criteria. Increasing amounts (in 10% steps as before) of questions according to this order were added in the measurements.

Figure 1 summarizes the results in terms of LAS for all the criteria we tested: random choice and likelihood based. As is typical in many active learning scenarios, random choice turned out not so easy to beat. Choosing sentences with the lowest likelihood was the best performing criterion when using about half of the test data, and was still better than random for lower percentages. Using the highest likelihood to parse questions provided the less improvements, consistently with the assumption that the parser already knew how to handle them and providing support that its estimates were indeed correct.

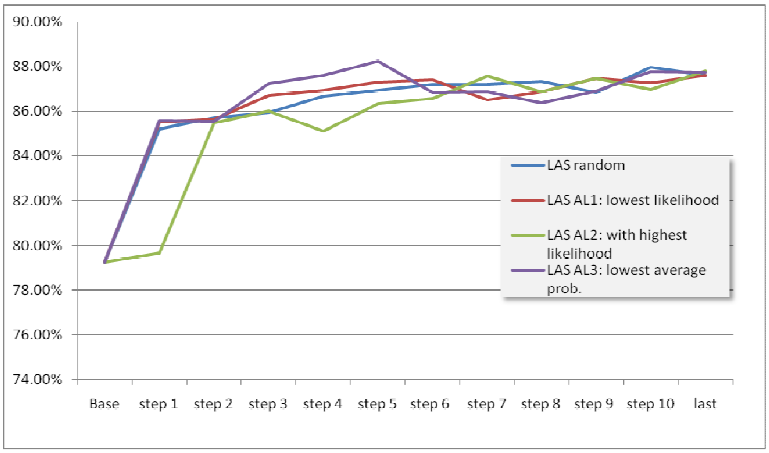

Figure 1. Learning curve with different selection criteria, including the random criterion

Active Learning Test

Once we figured out the best criterion for selecting questions for training, we tested its effectiveness by iterating the process according to the Active Learning metaphor. This is only an approximation of

[2] Half of the original Penn Treebank, excluding questions

a true Active Learning process, since we are using the same set of question from which to draw new examples, rather than a new set at each iteration.

At each iteration, a new parser is trained on the corpus produced in the previous iteration. So after step 1, the corpus is re-parsed with the new model and re-ordered before selecting the sentences for the next step on the active learning. As expected, the parsers improves very quickly: after a four steps we reach almost 87.53% LAS on questions (87% LAS on base) and using the full question corpus we reach 87.73% LAS. In fact it learns very little after the first couple of iterations, showing that indeed it learned most of what it could learn from the given set.

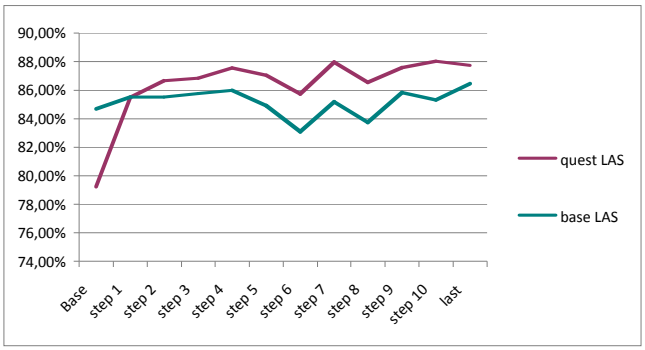

Figure 2. Result using lowest likelihood, re-ordering at each step.

Conclusions and Future Work

The experiments show than with a relatively small corpus (1100 questions) quite good accuracy can be obtained in parsing questions without hurting the performance on normal sentences. The availability of the resource we have built could be helpful to the NLP community not just for improving the accuracy of parsers but also in other high level natural language tasks which involve analyzing questions (e.g. question answering, dialog systems, etc).

We have also shown that different active learning methods can be explored in order to reduce the cost in building a question corpus (e.g. developing for other languages).

We will further investigate the use of more complex active learning techniques, specially the strategies that can be explored in order to build a corpus of questions in a semi-supervised way from unannotated texts.

Bibliography

1. [yanswers] Yahoo! Answers Comprehensive Questions and Answers version 1.0
2. [CoNLL] http://ifarm.nl/signll/conll/
3. [CHILDES] http://childes.psy.cmu.edu/
4. Ulf Hermjakob. 2001. Parsing and Question Classification for Question Answering. Proc. of the Workshop on Open-Domain Question Answering, ACL 2001.
5. [TREC QA] http://trec.nist.gov/
6. G. Attardi. Experiments with a Multilanguage non-projective dependency parser. In *Proc. of the Tenth CoNLL*, (2006).