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
E-mail: jordi@yahoo-inc.com, attardi@di.unipi.it, simi@di.unipi.it, hugoz@yahoo-inc.com

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.

1 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, most noticeably in Question Answering (Surdeanu, Ciaramita & Saragoza, 2008) or Frequently Asked Questions (FAQ) retrieval.

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 (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 CoNLL 2007 training corpus achieves an accuracy of approximately 86% Labelled Attachment Score (LAS) measured on our question test. This result is quite disappointing when compared to the 89% LAS accuracy that can be achieved with the same corpus in parsing regular sentences. 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 sentences in the corpus), 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?

We address these questions by means of *active learning*.

Active learning is a supervised machine learning technique in which the learner is allowed to choose the data from which it learns. An active learner generates *queries* for an *oracle* (e.g. a human annotator) to obtain labels for data instances selected from a larger set of unlabeled data. Active learning has been successfully applied in many modern machine learning problems where unlabeled data are abundant and easily obtained, but labelling is difficult, time-consuming, or expensive (Settles, 2010). In particular there is a growing interest in applying this technique to nearly all language technology tasks, as reported in a literature survey by Olsson (Olsson, 2009) and testified by the NAACL HLT 2009 Workshop on Active Learning for NLP (Ringger, Hertel & Tomanek, 2009).

The active learning process aims at reducing the human annotation effort, only asking for advice when the utility of the query is high. The primary question is therefore *query* formulation: how to choose which example (or examples) to try next.

There are many heuristics for choosing the examples: choosing examples where we don´t have data (Whitehead, 1991), where we perform poorly (Linden & Weber, 1993), where we have low confidence (Thrun & Möller, 1992; Donmez & Carbonell, 2008), where we expect it to

change our model (Cohn et al., 1990), and where we previously found data that resulted in learning (Schmidhuber & Storck, 1993).
Multi-classifier approaches use measures of disagreement among a committee of classifiers, obtained in different ways, as a measure of uncertainty (Freund et al., 1997).
A separate issue, which influences the speed and performance of the active learning process, is whether the learner should process a single instance or a batch of instances at each iteration. Adding one instance at a time slows the overall learning process down. If, on the other hand, a batch of instances is added, the learning progresses faster, but it becomes more difficult to find strategies for selecting a good batch. Metrics combining in various ways *informativeness* (inversely related to uncertainty), *representativeness* (related to density, computed with clustering techniques) and *diversity* (reducing repetitions) have been proposed to address this issue (Olsson, 2009). For example, in the context of statistical parsing, Tang et al. propose to cluster parsed sentences, represented as a series of parsing events, according to a similarity measure based on the Hamming distance[1]. A *representativeness* measure, based on the density of clusters, is then combined with a measure of uncertainty to form a selection criterion for sampling (Tang, Ruo & Roukos, 2002).
The optimal size of the batch is also a critical parameter, which needs to be tuned on the basis of the specific application.
Most of the empirical results in the published literature suggest that active learning works in practice, and *selective sampling* methods outperform *random sampling* (as typical in *passive* learning) in most learning task. This is often true even for simple query strategies, such as uncertainty sampling.

2 Question Corpus Construction

We collected unannotated questions[2] 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 (Yahoo! Answers) (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 CoNLL 2007 training corpus; then we corrected the PoS, parsed again with the correct Part of Speech and finally we manually revised the resulting dependency labels.

	Sentences	**Avg. sent. Length**	**Tokens**
Yahoo! Answers Corpus	800	11.35	9,080
TREC Corpus	500	7.5	3,750
Question Corpus	1,352	9.50	12,830

Table 1: Question 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.

3 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, by means of a suitable selection criterion, it chooses "interesting" examples from a non-annotated collection, so that it can be manually annotated and added to the training corpus for the next iteration. After labeling every pattern we re-compute interestedness of unlabeled points, choose the one with highest, label it, re-train, etc.
If the selection criterion is effective, a much smaller number of examples need 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 step is a single pattern. Adding more than one pattern at a time incurs in 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, when, for example, re-training takes a long time and we do not want human annotators 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 annotators 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.
In our case we decided that a batch of 100 questions at a time is a quite conservative addition to a comparatively much larger training corpus.

4 Testing selection criteria

The first series of experiments aimed at observing the effect of different selection criteria, compared to random sampling. In doing so, we expect to gain insights on the amount of data that it is necessary to achieve a satisfying performance.
Our experiments involved using a portion of the Penn Treebank (a random sample of sentences from the CoNLL

[1] The Hamming distance measures the number of substitutions required to turn a sequence into another.
[2] http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/

2007 English corpus, without questions, containing 250,805 tokens) as initial corpus of non question sentences (henceforth *base corpus*), the *question corpus* described above, and a further portion of the Yahoo! Answers as a source of unlabeled data.
Since we aim for a parser able to perform well on both questions and non-questions, we decided to monitor the accuracy of intermediate parsers in their ability to parse normal sentences, so as to ensure no significant decrease in overall performance while adapting to parse questions. For this reason the base corpus was randomly split between a *base training* and *base test*: the base training corpus contains 240,859 tokens, 9,946 sentences; the base test set 6,493 tokens, corresponding to 267 sentences.
The question corpus was randomly divided in 9,960 tokens (1,048 questions) for training and 2,539 tokens (252 questions) for test.

4.1 Random choice

The first selection criterion we tested is random choice. The experiments show the accuracy obtained by adding, to the *base training* corpus extracted from the CoNLL 2007 corpus, increasingly bigger subsets of randomly selected questions from the *question training* set. These subsets of increasing size were selected each time from the same pool using a different random seed. At each step the accuracy of the parser was measured on the *base test* set and on *question test* set. Accuracy is measured in terms of Labeled Attachment Score (LAS, the percentage of correctly attached and labeled tokens) and Unlabeled Attachment Score (UAS, the percentage of correctly attached tokens). With this experiment we meant to observe the effect of adding to the training set, batches of questions of variable size, with no specific selection criterion.
We repeated the experiment 5 times, using different seeds, in order to mitigate the effect of contingencies.

	base	100	200	300	400
quest LAS	77.20%	81.99%	83.54%	84.59%	85.22%
base LAS	84.69%	85.73%	84.88%	85.26%	85.34%

500	600	700	800	900	1000
85.10%	85.23%	85.92%	85.77%	85.81%	86.01%
85.56%	85.43%	85.32%	85.15%	85.49%	85.63%

Table 2: Results for the random choice selection

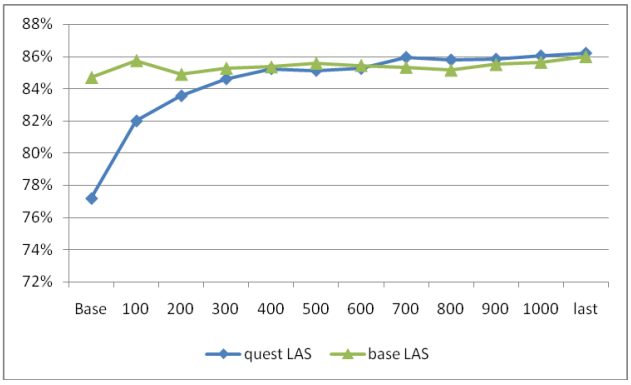

Figure 1: Random choice

Table 2 shows the average LAS scores for both the *base* and *question* (quest) test sets. The first column is the accuracy using just the *base* corpus with no questions at all, for training. The other columns report the results from adding to the base corpus 100 parsed questions each time (about 10%) from the question training set. The LAS reported is the average of the LAS obtained in the five repetitions of the experiment.
The baseline is a score of 77.20% LAS for the question test and 84.69% LAS for the non-question (base) test. These results were obtained with the DeSR parser (Attardi 2006). Other state-of-the-art parsers, such as the Malt parser (Nivre&Scholz, 2004), give similar results with these corpora[3]. This baseline accuracy is relatively low because in these experiments, for speeding up the runs, we traded up parsing accuracy for efficiency: in fact we used only about half of the CoNLL 2007 corpus for training, and reduced the number of iterations in building the model.
The results in Figure 1 show, on the average, 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. Adding only 600 questions the performance on the question test set gets even better than the performance on regular sentences. Figure 1 shows a plot of this experiment. The last step adds the residual 48 questions.
Even if this result is already very good, the following experiments aim at discovering better strategies for selective sampling, in order to see whether we can further reduce the effort needed to adapt the parser to questions.

4.2 Likelihood Estimates

We tested more sophisticated criteria to drive active learning based on likelihood estimates of a sentence parse. DesR is 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. Given a parsed sentence, the probability of each parsing step is therefore available to compute different metrics by which to estimate the confidence of the parser in its own output. For example:

a. *Likelihood of a parse tree*, computed as the product of the probabilities of all the steps used in building the tree;
b. *Average probability* of the parsing steps in building the tree;

In our experiments we selected sentences according to three different ordering criteria:

1. *Lowest likelihood* of sentence parse tree (LLK): aims at preferring sentences that were judged more difficult, by considering the likelihood of the parse tree;
2. *Highest likelihood* of sentence parse tree (HLK): prefers sentences that were judged easier by the parser, by considering the likelihood of the parse tree;
3. *Lowest average probability* (LAP): selects sentences that were judged more difficult by computing the average probability of each parsing step;
4. *Lowest normalized likelihood* (LNL): takes into

[3] Using the Malt parser with the default configuration we obtained a LAS score of 73.45% in parsing the question test set.

account the length of the parsed sentences by introducing a normalization factor (*likelihood*/*log*(*n*), where *n* is the number of tokens in the sentence).

The sentences in the question training corpus were parsed and then ordered a priori with these criteria. Increasing amounts (in 10% increments as before) of questions according to this order were added to the training corpus at each step and the performance of the parser evaluated on the base test and question test.

	base	100	200	300	400
RAND	77.20%	81.99%	83.54%	84.59%	85.22%
LLK	77.20%	82.87%	85.39%	85.19%	84.99%
HLK	77.20%	76.84%	77.79%	78.69%	80.19%
LAP	77.20%	82.71%	83.85%	84.80%	84.60%
LNL	77.20%	82.20%	85.47%	85.35%	84.17%

500	600	700	800	900	1000
85.10%	85.23%	85.92%	85.77%	85.81%	86.01%
85.58%	84.80%	85.58%	86.18%	87.12%	85.74%
82.99%	85.66%	84.29%	84.84%	84.48%	86.14%
86.10%	86.29%	86.33%	85.78%	86.10%	85.70%
85.66%	86.14%	85.19%	85.66%	85.98%	86.92%

Table 3: Results of evaluating criteria

Table 3 summarizes the results of parsing the question test set in terms of LAS, for all the criteria we tested: random choice (RAND) and three of the ordering criteria described above. Figure 2 (reported at the end of the paper) provides a direct comparison of the different selective sampling strategies.

As typical in many active learning scenarios, random choice turned out not so easy to beat. Choosing sentences with the lowest likelihood (LLK) was the best performing among the ordering criteria. Selecting new training examples according to their difficulty (estimated by the likelihood of the parse tree) helps the parser to learn faster than choosing sentences randomly. The normalized version of the likelihood measure (LNL) performs about the same as the non normalized version (and it is not reported in the figure), reflecting the fact that questions do not have a significant difference in length.

Moreover, adding only 200 questions to the training set, the accuracy on the question test set raises of more than 8% as opposed to 6% obtained by random sampling. Using the highest likelihood to select questions proves to be the worst choice, consistently with the assumption that the parser already knew how to handle easy cases. This also is an indication that the parser's estimates were indeed correct.

These results seems to indicate that it is quite easy to adapt a parser to handle questions, and that an active learning process needs only a few steps to produce a model with the desired accuracy. In our case, the accuracy in parsing questions easily outperforms that in parsing normal sentences.

5 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 paradigm. In fact the experiment is only an approximation of a true Active Learning process, since we are using the same, relatively small, set of questions from which to draw new examples, rather than a brand new set at each iteration.

At each step, a new parser is trained on the corpus produced in the previous iteration. So after step 1, the question training corpus is re-parsed with the new model and re-ordered, according to *lowest likelihood* in this case, before selecting the batch of sentences to add to the training corpus for the next active learning step. Figure 3 shows the learning curve obtained in ten active learning steps in comparison with random sampling, typical of passive learning.

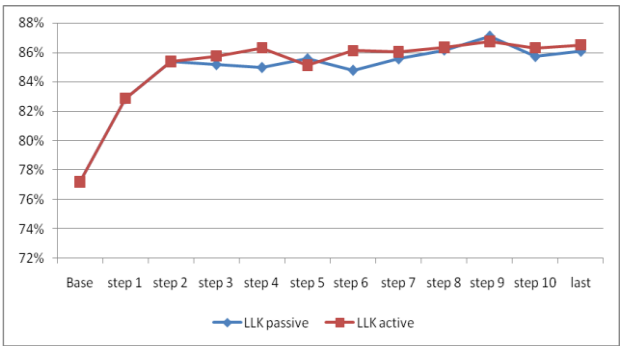

Figure 3: Comparison of active and passive learning

As expected, the parser improves very quickly: after four steps its accuracy reaches about 86.30% LAS on questions (less than 86% LAS on base); using the full question corpus the accuracy goes to 86.49% LAS. In fact it learns very little after the first four iterations, showing that indeed it learned most of what it could learn from the given set.

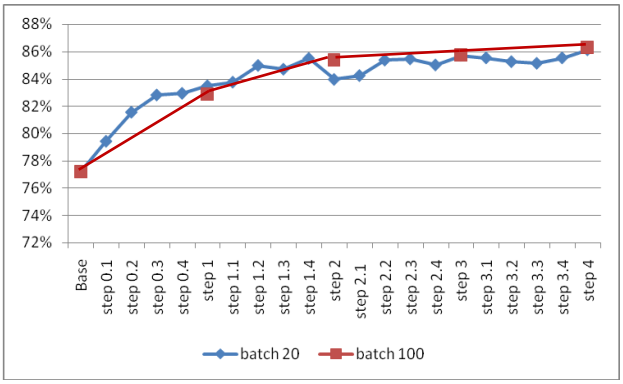

Figure 4: Active learning with different batch sizes

As a further experiment we wanted to observe whether smaller increments to the training corpus would have an effect on the learning curve: we tried with batches of 20 questions, instead of 100.

The results in Figure 4, show that, in this case, there is not much to be gained in retraining after small additions to the corpus.

6 Conclusions and Future Work

The experiments show than with a relatively small corpus (about 1000 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, Frequently Asked Questions retrieval, dialog systems, etc).

We have also shown that different active learning strategies, albeit very simple, can prove effective in reducing the cost in building a question corpus, e.g. developing such a corpus for other languages. In fact, we did build a similar corpus for Italian and the experience and outcomes were about the same.

We will further investigate the use of more elaborate active learning techniques, with the aim of building a larger and more complete corpus. We will especially consider the strategies that can be explored in order to build a corpus of questions in a semi-supervised way from unannotated texts.

7 Acknowledgments

This work has been supported in part by the Spanish Ministry of Education and Science, Torres Quevedo Programme.

8 References

Attardi, G. (2006). Experiments with a Multilanguage non-projective dependency parser. In *Proc. of the Tenth CoNLL*.

CHILDES. http://childes.psy.cmu.edu/

Cohn, D., Atlas, L., & Ladner, R. (1990). Training Connectionist Networks with Queries and Selective Sampling. In D. Touretzky (Ed.) *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann.

CoNLL. http://ifarm.nl/signll/conll/

Donmez, P. & Carbonell, J. G. (2008). Optimizing estimated loss reduction for active sampling in rank learning. International Conference on Machine Learning, ICML.

Freund, Y., Seung, S., Shamir, E. & Tishby, N. (1997). Selective sampling using the query-by-committee algorithm, *Machine Learning*, 28, pp. 133–168.

Hermjakob, U. (2001) Parsing and Question Classification for Question Answering. *Proc. of the Workshop on Open-Domain Question Answering*, ACL.

Linden, A. & Weber, F. (1993). Implementing inner drive by competence reflection. In H. Roitblat et al. (Eds.), *Proc. 2nd Int. Conf. on Simulation of Adaptive Behavior*, MIT Press, Cambridge.

Nivre, J. & Scholz, M. (2004). Deterministic Dependency Parsing of English Text. In *Proc. of COLING 2004*, Geneva, Switzerland, pp. 64–70.

Olsson, F. (2009). A literature survey of active machine learning in the context of natural language processing, Swedish Institute of Computer Science Technical Report, April 17.

Ringger, E., Hertel, R. & Tomanek, K. (Eds.) (2009). *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*.

Settles, B. (2010). Active Learning Literature Survey, Computer Science Technical Report 1648, University of Wisconsin-Madison, Jan. 26.

Schmidhuber, J. & Storck, J. (1993). Reinforcement driven information acquisition in nondeterministic environments. Tech. Report, Fakultat Informatik, Technische Universitat München.

Surdeanu, M., Ciaramita, M. & Zaragoza, H. (2008). Learning to Rank Answers on Large Online QA Collections, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics.

Tang, M., Luo, X. & Roukos, S. (2002). Active Learning for Natural Language Parsing, *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 120–127.

Thrun, S. & Möller, K. (1992). Active exploration in dynamic environments. In J. Moody et al. (Eds.) *Advances in Neural Information Processing Systems 4*. Morgan Kaufmann.

TREC QA. http://trec.nist.gov/

Yahoo!Answers. Yahoo! Answers Comprehensive Questions and Answers version 1.0

Webscope. http://webscope.sandbox.yahoo.com/

Whitehead, S. (1991). A study of cooperative mechanisms for faster reinforcement learning. TR-365, Dept. of Computer Science, Rochester Univ., Rochester, NY.

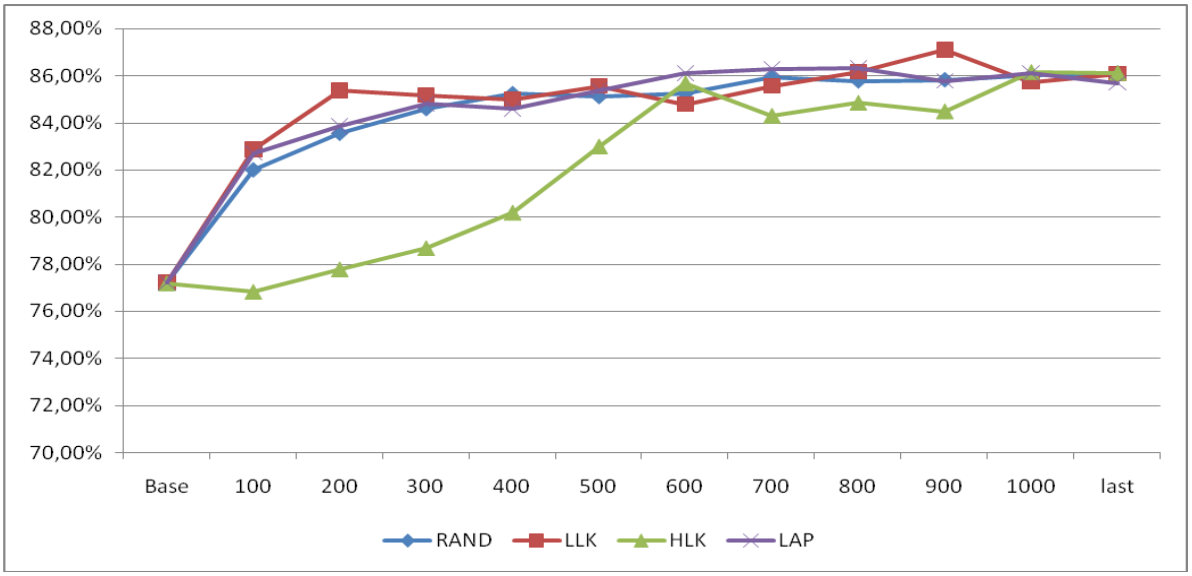

Figure 2: Comparing different selection criteria