A Wikipedia based Algorithm for Online Adaptation of a Syntactic Parser

Gourab Kundu
University of Illinois, Urbana-Champaign
Urbana, IL. 61801
kundu2@illinois.edu

Abstract

Adaptation of models to a new domain is important for many natural language tasks. Because without adaptation, NLP tools trained on news domain achieve sub par results on other domains. In many practical scenarios, the identity of the domain of the test set is unknown. For this difficult but important setting, we propose a novel method of adaptation using entity disambiguation systems to Wikipedia. We get significant improvements for adapting a syntactic parser trained on news domain to biomedical domain.

1 Introduction

Traditionally, annotations for natural language processing tasks like POS tagging, shallow parsing, syntactic parsing etc. have been done on a corpus of articles from Wall Street Journal. Statistical models for these tasks have mostly been trained and tested on text from this domain. As a result, the state-of-the-art models perform very well on news text but their performance drops significantly when tested on a different domain.

Self-training has been used successfully for adapting syntactic parsers (McClosky et al., 2006b). In this method, to adapt a parser trained on WSJ domain to a new domain, the parser is first applied to millions of sentences in the new domain to produce parse trees automatically. These automatically labeled data from the new domain are then added to the manually labeled data from the WSJ domain and the parser is retrained from this combined labeled data. However, this technique assumes that the domain of the test set is known and necessitates that the researcher collects significant amount of unlabeled data for this domain. This assumption does not hold in online scenarios where inputs arrive in small chunks (may be even one sentence at a time) from diverse domains and it is infeasible to manually identify the domain of the input sentence and collect clean unlabeled data. An example of this scenario is the real-time online parsing demo systems which are used by users from different domains. It is often the case that users provide *short* text from *arbitrary domains* as input. When parsing the world wide web, similar situations arise since many web sites contain small amounts of text and each web site can be regarded as a separate domain. As a result, training a model for each web site may not be feasible since sufficient text is not readily available and collecting similar text is not trivial.

In this paper, we propose *OSPA* (Online Syntactic Parser Adaptation). This is different than traditional adaptation settings since in this setting, each time the parser is given a single sentence as input and the domain of the input sentence is assumed unknown. Our goal is to adapt to the test sentence *on the fly*. There are two essential steps to adapt to the test sentence:

- Significant amount of text related to the test sentence needs to be collected.
- The parser needs to be adapted to the test sentence based on the collected text.

We propose to solve the first problem by annotating

the test sentence with an entity disambiguation system to Wikipedia (Ratinov et al., 2011; Cucerzan, 2007). We crawl text from the Wikipedia pages of entities that the disambiguation system identifies in the test sentence. Then, a naive way for solving the second problem is to perform self-training, i.e., run the syntactic parser on the collected text and then retrain the parser on this annotated text. However, this might be computationally prohibitive since the processing time for each sentence will be dominated by the training time of the syntactic parser which is very large. We propose a novel algorithm OSPA where retraining the parser can be avoided. Instead, for each unknown word, we infer the label of the closest phrase encompassing that word by a majority voting scheme from the output of the parser on the collected text. Our experiments on adaptation of Charniak parser (Charniak and Johnson, 2005) from WSJ domain to biomedical domain demonstrate significant improvements. Overall, our contributions are:

- We propose a way to collect text related to test sentence using entity disambiguation system.
- We propose a novel algorithm for adapting syntactic parser in an efficient manner that does not need retraining the parser.

2 Related Work

There has been previous studies in the literature investigating how a parser trained on one domain performs when tested on a different domain. (Gildea, 2001) shows that if we add even small amount of in-domain treebank to a large amount of out-of-domain treebank, the parser performs much better than the parser trained on out-of-domain treebank only. (Roark and Bacchiani, 2003) combined the in-domain treebank with out-of-domain treebanks in a maximum a posteriori framework such that the parser trained on this sophisticated mixing performs better than parser trained on vanilla mixing. (Roark and Bacchiani, 2003) defined a prior on the PCFG rule probabilities based on the probability of that rule estimated using the out-of-domain treebank. Using this prior and counts from in-domain treebanks, the grammar that has the maximum posteriori probability was estimated. They defined the prior in a way such that the maximum a posterior estimation turns out to be a merging of scaled counts of PCFG rules from in-domain and out-of-domain treebanks. The scaling parameter is tuned over an in-domain held out set. In one adaptation setting (which they call "unsupervised adaptation"), no in-domain treebank was used except the held out set. The parser trained on out-of-domain treebanks is used to parse large amounts of text from in-domain and the counts of rules from these parses are merged with counts from out-of-domain treebanks. In their reported results, they used a value for the scaling parameter that was tuned on in-domain held out set. So even in this unsupervised setting, they actually need an in-domain set for tuning the scaling parameter.

The difference between (Gildea, 2001) and our work is that in our work, we have not used any in-domain treebanks. We assume that the domain of the input sentence is not even known. The unsupervised adaptation setting of (Roark and Bacchiani, 2003) is similar to ours except our work focuses on streaming scenarios where text comes in small chunks. So we cannot have an in-domain annotated set for tuning or learning. Moreover, (Roark and Bacchiani, 2003) collected raw text manually using the domain information of the test set but in our case, the domain of the test set is unknown and searching relevant raw text is part of our problem setting. Moreover, in our adaptation scheme, we do not need to adjust the counts of the parser, instead, for an input sentence, we select the most likely parse that satisfies as many constraints as possible. So it can be thought of as "Adaptation without Retraining".

Recently self-training has emerged as a popular technique for parser adaptation. (McClosky et al., 2006a) showed that if a parser trained on the Wall Street Journal corpus is self trained on NANC (North American News Corpus), the parser's performance improves significantly. (McClosky et al., 2006a; McClosky et al., 2006b) showed that the performance of the parser improves for both news domain and fiction domain test sets. (McClosky and Charniak, 2008) built a parser for biomedical domain by self-training on the biomedical domain instead of NANC. Our setting is different from (McClosky and Charniak, 2008) since we work in a streaming setting where collection of raw text is part

of the problem and retraining the parser over collected text for each sentence is computationally infeasible.

Perhaps the closest setting to our work is (McClosky et al., 2010). In that work, the authors build parsing models from annotated data of multiple domains and at test time, given some text from an unknown domain, propose a linear combination of the different models. However, in their setting, the text from the unknown domain was significantly large compared to our setting of short input text and we use annotated data from only one domain (Wall Street Journal).

Adaptation in streaming scenarios is relatively new in the literature. In (Umansky-Pesin et al., 2010), adaptation of POS tagger from WSJ to biomedical domain was done using web queries. Given an input sentence, queries were constructed, similar sentences were collected from the web and the input sentence was tagged based on the analysis on the collected text. (Rüd et al., 2011) addressed named entity tagging in queries. They included as features the titles, URLs of results returned by a web search engine for each query. (Ganchev et al., 2012) adapted POS tagger from full sentences to queries. Compared to these works, we do not use a web search engine. Repeated queries to a search engine and extraction of text from html pages are difficult and time consuming. Commercial search engines put a limit to the number of allowed queries per day using the search engine API. We use Wikipedia which is a free resource and the articles follow a consistent format.

3 Motivating Example

Figure 1 shows two subtrees of the predicted parse tree for the following sentence:

> Rather , IFN-gamma antagonized the effect of IL-4 and suppressed the DC and MGC formation induced by GM-CSF + IL-4 and M-CSF + IL-4 , respectively .

In two out of the three mentions of "IL-4", the parser made a mistake by identifying it with the phrase type "NX" instead of "NP". To find out the correct phrase type of "IL-4", we need to observe its usage in more text which is unavailable in our setting.

Algorithm 1 Online Syntactic Parser Adaptation (OSPA)

1: Input: Test Sentence t, Parser M trained on news domain
2: Output: Parse tree P for t
3: Collect raw text using Wikipedia
4: Parse the collected raw text using M
5: **for** each word w in t that was unseen in training domain **do**
6: Find the most frequent base phrase type of w from the parsed raw text
7: **end for**
8: Use M to generate the top K parses T for t
9: **for** each parse P in T **do**
10: calculate the number of violations in P
11: **end for**
12: select the set of parses that have the minimum violation V
13: select the parse from V that has the highest probability from M

However, an entity disambiguation system (Wikifier) can correctly disambiguate mentions like "IL-4", "GM-CSF" and "M-CSF" to their corresponding entities in the Wikipedia. From these Wikipedia articles, we can collect the raw text. Table 1 shows some example sentences collected from the Wikipedia article "Interleukin-4", the article into which the Wikifier disambiguates the word "IL-4". Charniak parser correctly predicts "IL-4" as a noun phrase in all these sentences.

The cell that initially produces IL-4, thus inducing Th0 differentiation, has not been identified.
Overproduction of IL-4 is associated with allergies.

Table 1: Example sentences containing "IL-4" from Wikipedia article "Interleukin-4"

4 Algorithm

At first, raw text is collected from Wikipedia using Wikifier and then parsed using the parser trained on WSJ domain (Lines $3-4$). For each word in the test sentence that was unseen in the training domain, the most frequent base phrase type (immediate par-

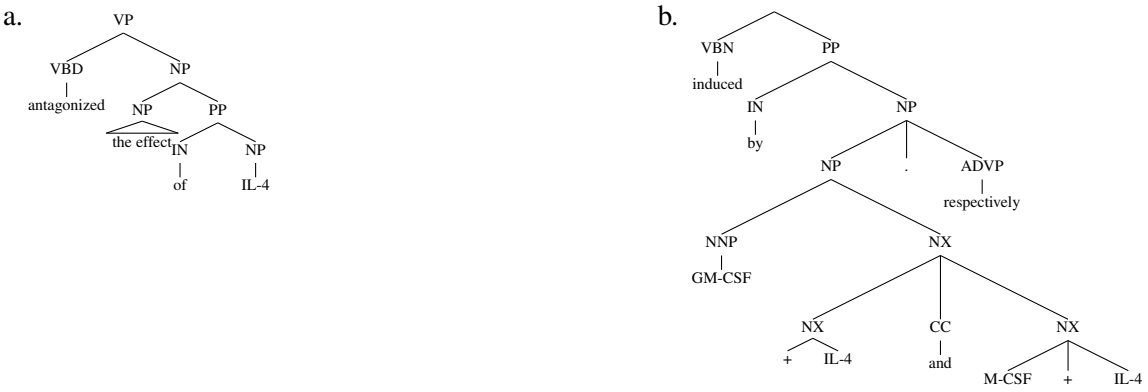

Figure 1: Correct (a) and incorrect (b) fragments of a parse tree predicted by the parser from (Charniak and Johnson, 2005)

ent of the POS tag in the parse tree) for that word (Lines $5-7$) is found from the parsed Wikipedia text. For each parse in the top K parses for the test sentence, the total number of violations is calculated. A violation occurs when an unknown word is assigned a base phrase type different than the majority phrase type (Lines $8-11$). Finally the parse tree with the minimum violation is selected. If there are multiple parses that have the minimum violation, the parse tree that was ranked the highest by the parser is selected. (Lines $12-13$).

5 Experiments

We use the parser from (Charniak and Johnson, 2005). This parser is trained on sections $2-21$ of Penn treebank. The test domain is biomedical domain. For this, we use the Genia treebank (Tateisi et al., 2005). The number of sentences in WSJ is roughly $40,000$. We follow the division of Genia treebank as in (McClosky and Charniak, 2008). We test on the first 1000 sentences of the development section of Genia treebank from the division of (McClosky and Charniak, 2008).

Our evaluation criteria is the labeled bracketing F1 measure. From Table 2, it is seen that OSPA improves over the baseline. The baseline is directly applying the WSJ trained parser on biomedical data. F1 improvement is 0.4% which is significant given that our experimental setting is very hard where we do not have any prior knowledge or raw text for each sentence and we cannot perform retraining.

Table 3 shows the labeled bracketing F1 measure for both baseline and OSPA for several common phrase categories. It shows that OSPA improves over most of the common phrase types.

Method	Precision	Recall	F1
baseline	70.73	79.92	75.04
OSPA	70.98	80.47	**75.44**

Table 2: Results on WSJ to biomedical adaptation

Phrase Type	baseline (F1)	OSPA (F1)
SBAR	86.5	**87.0**
NP	74.3	**74.7**
VP	85.5	**86.0**
PP	74.4	**74.7**
S	69.3	**69.5**

Table 3: Results on WSJ to biomedical adaptation

6 Future Work

OSPA can be thought of as an algorithm for finding the configuration that satisfies the highest number of constraints. The only constraint used was: the base phrase type of each unseen word must be the most frequent base phrase type over the parsed corpus. In future, we want to experiment with different types of constraints based on the Wikifier output.

7 Conclusion

In many practical scenarios, the domain of the test sentence is unknown. Adaptation in these scenarios is very difficult. For these scenarios, we propose a simple algorithm that gives significant improvements over a state-of-the-art parser.

References

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *ACL*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference of EMNLP-CoNLL*.

Kuzman Ganchev, Keith Hall, Ryan McDonald, and Slav Petrov. 2012. Using search-logs to improve query tagging. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 238–242. Association for Computational Linguistics.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 167–202.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 101–104. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006a. Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 337–344. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 28–36. Association for Computational Linguistics.

L. Ratinov, D. Downey, M. Anderson, and D. Roth. 2011. Local and global algorithms for disambiguation to wikipedia. In *ACL*.

Brian Roark and Michiel Bacchiani. 2003. Supervised and unsupervised pcfg adaptation to novel domains. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 126–133. Association for Computational Linguistics.

Stefan Rüd, Massimiliano Ciaramita, Jens Müller, and Hinrich Schütze. 2011. Piggyback: Using search engines for robust cross-domain named entity recognition. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 965–975.

Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Junichi Tsujii. 2005. Syntax annotation for the genia corpus. In *Proceedings of IJCNLP*, volume 5, pages 222–227.

Shulamit Umansky-Pesin, Roi Reichart, and Ari Rappoport. 2010. A multi-domain web-based algorithm for pos tagging of unknown words. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1274–1282. Association for Computational Linguistics.