Leveraging Lexical Substitution for Parser Adaptation

Spencer Caplan, Neal Fox
Brown Laboratory for Linguistic Information Processing (BLLIP)
Brown University
Providence, RI, 02912
{scaplan | npfox}@cs.brown.edu

David McClosky
IBM Research
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598
dmcclosky@us.ibm.com

Eugene Charniak
Brown Laboratory for Linguistic Information Processing (BLLIP)
Brown University
Providence, RI, 02912
ec@cs.brown.edu

Abstract

When a statistical parser is tested on text from a different genre than the data on which it was trained, its performance suffers. Adaptive parsing methods aim to reduce this decrement. We investigate the efficacy of lexical substitution (LS) for parser adaptation. We hypothesize that replacing unknown (i.e., undertrained) words with similarly distributed known words will allow the parser to use syntactically-relevant lexicalized information that it has trained on. Our LS model leverages both implicit syntactic information (word co-occurrence statistics) and explicit syntactic information (similar words' POS distributions) to identify appropriate substitutions. Our results show that LS improves performance of the WSJ-trained BLLIP parser (Charniak, 2000) for three out-of-domain corpora. Importantly, LS does not hurt parsing for in-domain text. Our results suggest promising directions for future work on parser adaptation.

1 Introduction

Modern statistical parsers rely on large, painstakingly hand-annotated corpora in order to train their parameters. However, because patterns of language vary between genres, parser performance suffers when training and testing sets are drawn from different domains (Sekine, 1997; Gildea, 2001; Bacchiani et al, 2006; McClosky et al, 2006b).

Unfortunately, in practical applications, in-domain "gold standard" labeled data is often unavailable and would be prohibitively expensive and time-consuming to collect. Therefore, developing methods of *parser adaptation* has the potential to enhance the usefulness of standard parsers for applications to non-news data. Several such methods have been discussed in the literature, including self-training (McClosky et al, 2006b), co-training (Steedman et al, 2003), re-training with lexical information (e.g., named entities, POS tags; Lease et al, 2005; Rimell & Clark, 2008), and parse-reranking with web-scale features (Bansal & Klein, 2011). Here, we propose another technique: lexical substitution (LS).

2 Unknown Words and Lexical Similarity

There are many reasons why parsing performance might suffer when training and testing data come from different domains. Lease et al (2005) observe that parsers encounter unknown words at a higher rate when evaluating on out-of-domain text. Thus, to the extent that a parser relies on lexicalized information, it will have less information in novel domains with more dissimilar lexicons. We propose that replacing *unknown* words with *known* words that behave similarly will allow the parser to leverage the wide array of lexicalized information it has trained on.

In order to find known words that would be suitable replacements for unknown words, we implement a model of lexical-semantic similarity that quantifies the degree to which two words' bigram distributions are similar. We hypothesize that if an unknown word (w_1) and a known word (w_2) tend to appear before and after the same context words with similar relative frequencies, then replacing w_1 with w_2 will enhance parser performance.

Many models of lexical similarity rely on word co-occurrence statistics (Riordan & Jones, 2010;

Weeds et al, 2004). For instance, Latent Semantic Analysis (LSA; Landauer & Dumais, 1997) measures how often words co-occur in the same paragraph. However, LSA is not designed to capture words' syntactic roles – it is more likely to deem *Sudanese* and *refugee* similar than *Sudanese* and *Norweigian*. Zhao et al (2011), on the other hand, present a model that evaluates word similarity based on how often words occur in the same bigram contexts. They suggest that clustering of these vectors successfully groups words by POS tag. We apply a similar technique to find a known word that not only has the same POS tag as an unknown word, but might also offer hints to a parser about how that unknown word is likely to interact with other constituents in a sentence.

3 Lexical Substitution Model

In order to find a maximally-similar, sufficiently trained substitute for a word in some test corpus (TEST) about which a parser has little knowledge, several issues must be considered. Chief among them are: (1) how should we quantify two words' similarity, (2) how should we apply that metric to select the best replacement word, (3) under what conditions should a word in TEST be replaced or a word in TRAIN considered as a potential replacement? We examine each of these questions in describing our model of LS and its application to cross-domain parsing. For the sake of notation, recall that we are seeking pairs of related words (w_q, w_c) that match a trained word (w_c: *candidate* replacement) and an untrained word (w_q: *query* word we want to replace). Together, the *query* words (Q) and the *candidate* words (C) compose what we call the model's *critical words*.

3.1 Distributional Divergence

Because we need to match trained words to untrained words, a linking corpus (LC) that includes both is needed. From LC, we compute each critical word's (w_i) bigram distribution by counting the number of times w_i occurs before and after each word c_x that we consider as bigram contexts. Thus, for each w_i, if there are N context words ($c_1...c_N$), a definition vector $d[w_i]$ of length $(N+1)\times 2$ is calculated. $d[w_i]$ has a component indexing the count of occurrences of bigram $[w_i\ c_x]$ and of bigram $[c_x\ w_i]$ for all $x \leq N$, plus a pair of components summing the same counts over all words not in $c_1...c_N$. $d[w_i]$ is then smoothed by adding α to the count of bigrams which never occurred in LC. Finally, $d[w_i]$ is normalized, yielding a probability distribution.

For each query word w_q encountered by the parser in TEST, our task is to select a replacement w_c that the parser would better recognize. Thus, to determine the most similar w_c for a given w_q, we compute the KL-divergence between each $d[w_c]$ and that $d[w_q]$. This comparison tells us how much information about w_q's contextual distribution would be lost by replacing all instances of w_q in LC with w_c. We call $D_{KL}(d[w_q]||d[w_c])$ the *distributional divergence* between w_q and w_c ($DD[w_q,w_c]$).

In calculating distributional divergence (DD), there are a number of important implementation considerations. Our publicly available software package[1] allows users to manipulate many relevant parameters, including N. For the present experiments, the set of context words ($c_1...c_N$) was the union of Q, C, and the 6,000 most common unigrams in the 2009 English Google Books corpus (GB; Michel et al, 2010).

Another parameter users can manipulate is α, which can be interpreted as indexing our confidence that bigrams in our LC are representative of language use in TRAIN/TEST (higher values indicate more uncertainty). The present experiments employed GB (bigrams) as the LC and we tuned α via grid-search (α = 0.01, 0.1, 1; see **4.1**).

3.2 Candidate Reranking

We could simply select the w_c with the smallest $DD[w_q,w_c]$ as the best replacement for w_q. However, because most words have multiple senses, it is unavoidable that replacing w_q with any w_c could bias the parser with lexical idiosyncrasies specific to w_c. Our goal is to select a w_c that is highly related to w_q, while avoiding such biases.

Notably, for a large enough C, several w_c will be highly related to a given w_q (i.e., have a low $DD[w_q,w_c]$). We selected a subset $C_M \subset C$ consisting of the M candidates with the lowest $DD[w_q,w_c]$. We arbitrarily set M=10 ($C_M = C_{10}$). With C_{10}, we then reordered the potential replacements using an algorithm we call *Candidate Reranking*. Candidate Reranking (CR) leverages explicit information about the POS-tags of each w_c in C_{10} so as to avoid selecting a w_c that would mislead the parser.

[1] Link to software to be included pending anonymous review

Because the parser has not trained on w_q, it does not have access to information about which POS-tags ($t \in T$) w_q tends to take on (its *POS-distribution* $P(T|w_q)$). It does, however, have information about $P(T|w_c)$ for each w_c in C_{10}. We compute an average POS-distribution $P^*(T|w_c)$ across C_{10}, effectively canceling out the lexical idiosyncrasies of any individual w_c and yielding an estimate of $P(T|w_q)$ (Eq. 1). Finally, we rerank the elements of C_{10} based on their similarity to $P^*(T|w_c)$, selecting the most similar (Eq. 2). We further explore CR during model development (see **4.1**).

$$P(T \mid w_q) \approx P^*(T \mid w_c) = \frac{1}{C_M} \sum_{i=1}^{|C_M|} n_{w_i} P(T \mid w_i) \quad \text{(Eq. 1)}$$

$$\arg\min_{w_c} KL(P^*(T \mid w_c) \| P(T \mid w_c)) \quad \text{(Eq. 2)}$$

3.3 Query and Candidate Word Criteria

We have largely motivated the use of lexical substitutions by appealing to uncertainty that plagues a trained parser when it encounters unknown words. However, it is unclear how rare a w_q should be in TRAIN to warrant being replaced if/when encountered in TEST. For instance, if a lexical item is seen twice (instead of 0 times) in TRAIN, then replacing it with a better-trained w_c may provide a stronger signal to the parser than leaving the "attested-but-under-trained" w_q. During tuning, we examined several criterion levels for f_Q (i.e., the number of times some w_q in TEST must appear in TRAIN to *not* be replaced, considering $f_Q = 1, 3, 5, 7$).

Conversely, for some w_c in TRAIN, how frequent should it be in order to warrant being considered as a possible replacement for each w_q? During tuning, we also examined several criterion levels for f_C (i.e., the number of times some w_c must appear in TRAIN to be considered as a potential replacement). Note that if $f_C < f_Q$, then a w_q could be replaced by itself since $DD[w_i,w_i] = 0$. However, the parser could be confident enough about some w_i to exclude it from Q without being confident enough to include it in C. Therefore, we considered $f_C = 3, 5, 7$ during tuning for $f_C \geq f_Q$.

Finally, our definition model $d[w_i]$ is only as good as the information available in the LC. Thus, $d[w_i]$ for a very rare w_i might not prove useful, and could lead us to make poor replacements. Therefore, we also tuned f_{GB} (i.e., the number of times a word must appear in GB to be included in Q or C, considering $f_{GB} = 40, 1e4, 5e4, 1e5, 1e6$).[2]

	Baseline	LS	LS + CR
POS %	85.3	86.4	89.4
f-score	83.9	84.1	84.5

Table 1: POS-tag accuracy for query words in $\text{Brown}_{\text{tune}}$ and overall parsing *f*-score: without lexical substitution (Baseline), with substitution but no candidate reranking (LS), and with reranked substitutions (LS + CR).

4 Experiments and Results

We investigated the effect of lexical substitution on cross-domain parsing using the publicly available[3] BLLIP parser (Charniak, 2000) trained on sections 02-21 of the Wall Street Journal ($\text{WSJ}_{\text{train}}$) portion of the Penn Treebank (Marcus et al, 1993).

4.1 Development

We used the development section of the Brown corpus ($\text{BROWN}_{\text{tune}}$; Kucera & Francis, 1967) to tune four parameters (α, f_Q, f_C, f_{GB}) in a 3×4×3×5 grid-search. The parameter settings that yielded the largest overall improvement in *f*-score were then used for all subsequent development and testing.[4]

To explore the effectiveness of CR we parsed the development section of the Brown corpus ($\text{BROWN}_{\text{tune}}$; Kucera & Francis, 1967) three times: without substitutions (Baseline), with substitutions based only on DD (LS), and with reranked substitutions (LS + CR).[5] Table 1 shows that reranking the ten most similar candidates prior to LS improves on the model's overall parsing *f*-score. There is a corresponding gain in the accuracy of assigning POS-tags to unknown words during parsing. This highlights the efficacy of explicitly leveraging syntactic information during CR, complementing the implicit syntactic information extracted by *DD*.

4.2 Testing

Following the development phase, we tested the extent to which lexical substitution could improve parsing in four corpora: the test section of the

[2] f_{GB} evaluated on unigrams; f_{GB}<40 are excluded from GB
[3] http://www.github.com/BLLIP/blip-parser
[4] The optimum parameters selected following grid-search were: α=1, f_Q=5, f_C=7, f_{GB}=5e4. All lexical substitutions during this tuning phase were selected following CR.
[5] Lexical substitutions in the CR experiments described were made based on optimum parameter settings from tuning.

	Evaluating on all sentences			*Evaluating on sentences with changed parses*		
	# sentences	**Baseline**	**Baseline + LS**	**# sentences**	**Baseline**	**Baseline + LS**
$BROWN_{tune}$ (best)	2078	83.9	**84.5 (p<0.01)**	684 (of 1502)	79.6	**81.0 (p<4e-5)**
$BROWN_{test}$	2425	84.1	**84.4 (p=0.07)**	799 (of 1715)	80.5	**81.2 (p=0.07)**
QB	4000	85.6	**85.8 (p=0.07)**	825 (of 2595)	81.6	**82.5 (p=0.07)**
BNC	999	82.5	82.8	576 (of 959)	79.9	80.3
WSJ_{test}	2416	89.7	89.7	332 (of 1160)	85.6	85.2

Table 2: *f*-score performance for the WSJ-trained BLLIP parser (Charniak, 2000), with and without lexical substitution (LS), evaluated on all sentences and on sentences where LS changed the parse. $BROWN_{tune}$ results (shaded) with LS reflect the highest *f*-score achieved during grid-search. Bolded results indicate experiments where LS significantly improved parsing. LS never significantly harmed performance. When evaluating on sentences with changed parses, *# sentences* is listed as: *# sentences with changed parses (of # sentences with replacements)*.

Brown corpus ($BROWN_{test}$); Question Bank (QB; Judge et al., 2006); the British National Corpus (BNC; Foster & Genabith, 2008); and section 23 of WSJ (WSJ_{test}). Results appear in Table 2.[6] Parsing *f*-scores were evaluated for all sentences in each test corpus and on sentences for which substitutions resulted in changes to the parse.

Overall, *f*-score performance was improved by lexical substitution (LS) in two of the test corpora ($BROWN_{test}$, QB). In a third test corpus (BNC), improvements were not statistically significant, although the raw f-score improvement with LS was consistent with the other corpora (0.3%); notably, BNC is the smallest of the test corpora with only 999 sentences. To further investigate the quality of our lexical substitutions, we examined only sentences in which making substitutions altered the parse. As is clear from the lower overall baseline of these changed parses, lexical substitutions tended to alter the sentences with the worst parses. LS had no impact (positive or negative) on parsing performance in an in-domain test corpus (WSJ_{test}).

5 Discussion and Conclusion

In general, the present work suggests that LS is a promising direction for further research into cross-domain parsing and parser adaptation. As a first exploration of this technique, we have shown that LS can improve parsing of text in a novel (non-news) domain. Interestingly, this boost was evident in corpora with very different characteristics (from each other). For instance, the syntactic trees a parser encounters in QB (namely, questions) are highly dissimilar from the trees in either WSJ_{train} or in the other test corpora. The uniform improvements suggest that LS may prove to be a broadly applicable technique for parsing. That said, it raises questions about the conditions under which LS might improve parsing most, which warrants future research.

For one, while LS achieves significant improvements for cross-domain parsing in the baseline Charniak parser, future work should consider whether or when LS might improve over a state-of-the-art parser (e.g., the self-trained Charniak parser with discriminative reranking; McClosky et al, 2006a). It is possible that adapting the LS model to consider the full vocabulary of a self-trained parsing model could provide even further boosts to performance.

It should be no surprise that LS does not improve parsing results for the in-domain test corpus, (WSJ_{test}). Unknown words are much rarer in WSJ_{test}, and patterns of syntactic usage are obviously much more similar for in-domain testing. Moreover, as baseline performance is higher, any improvements would inevitably be smaller. However, it is notable that in no case did LS significantly hurt parsing performance. This is important because it suggests that, for many practical applications, allowing for judicious word replacements has the potential to improve parser performance, while carrying very little risk.

Without the luxury of gold-standard data for training in many domains, adaptive parsing techniques will only become more important for future natural language processing work. Lexical substitution represents a promising direction for future research that could make high-performance parsers more dependable for cross-domain parsing.

Acknowledgments

To be added after anonymous review to NAACL.

[6] Significance was evaluated with random permutation testing

References

Bacchiani, M., Riley, M., Roark, B., & Sproat, R. (2006). MAP adaptation of stochastic grammars. *Computer speech & language*, *20*(1), 41-68.

Bansal, M., & Klein, D. (2011, June). Web-scale features for full-scale parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (pp. 693-702). Association for Computational Linguistics.

Charniak, E. (2000, April). A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference* (pp. 132-139). Association for Computational Linguistics.

Gildea, D. (2001). Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*(pp. 167-202).

Judge, J., Cahill, A., & Van Genabith, J. (2006, July). Questionbank: Creating a corpus of parse-annotated questions. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* (pp. 497-504). Association for Computational Linguistics.

Francis, W. N., & Kucera, H. (1979). Brown Corpus manual: Manual of information to accompany a standard corpus of present-day edited American English for use with digital computers. *Brown University, Providence, Rhode Island, USA*.

Foster, J., & Van Genabith, J. (2008). Parser evaluation and the bnc: Evaluating 4 constituency parsers with 3 metrics.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*(2), 211.

Lease, M., & Charniak, E. (2005). Parsing biomedical literature. In *Natural Language Processing–IJCNLP 2005* (pp. 58-69). Springer Berlin Heidelberg.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*,*19*(2), 313-330.

McClosky, D., Charniak, E., & Johnson, M. (2006a, June). 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* (pp. 152-159). Association for Computational Linguistics.

McClosky, D., Charniak, E., & Johnson, M. (2006b, July). 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* (pp. 337-344). Association for Computational Linguistics.

Michel, J. B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Pickett, J. P., ... & Aiden, E. L. (2011). Quantitative analysis of culture using millions of digitized books. *science*, *331*(6014), 176-182.

Rimell, L., & Clark, S. (2008, October). Adapting a lexicalized-grammar parser to contrasting domains. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 475-484). Association for Computational Linguistics.

Riordan, B., & Jones, M. N. (2011). Redundancy in Perceptual and Linguistic Experience: Comparing Feature□Based and Distributional Models of Semantic Representation. *Topics in Cognitive Science*, *3*(2), 303-345.

Sekine, S. (1997, March). The domain dependence of parsing. In *Proceedings of the fifth conference on Applied natural language processing* (pp. 96-102). Association for Computational Linguistics.

Steedman, M., Baker, S., Crim, J., Clark, S., Hockenmaier, J., Hwa, R., ... & Sarkar, A. (2003). *CLSP WS-02 final report: Semi-supervised training for statistical parsing*. Technical report, Johns Hopkins University.