Parsing with Subdomain Instance Weighting from Raw Corpora

Barbara Plank[1], *Khalil Sima'an*[2]

[1]Alfa informatica, Faculty of Arts, University of Groningen, The Netherlands
[2]Language and Computation, Faculty of Science, University of Amsterdam, The Netherlands
b.plank@rug.nl, k.simaan@uva.nl

Abstract

The treebanks that are used for training statistical parsers consist of hand-parsed sentences from a single source/domain like newspaper text. However, newspaper text concerns different subdomains of language use (e.g. finance, sports, politics, music), which implies that the statistics gathered by generative statistical parsers are averages over subdomain statistics. In this paper we explore a method, subdomain instance-weighting, that exploits raw subdomain corpora for introducing subdomain statistics into a state-of-the-art generative parser. We employ instance-weighting for creating an ensemble of subdomain specific versions of the parser, and explore methods for amalgamating their predictions. Our experiments show that subdomain statistics extracted from raw corpora can even improve the quality of the n-best lists of a formidable, state-of-the-art parser.
Index Terms: statistical parsing, adaptation, subdomains, instance weighting

1. Motivation

Generative models for statistical parsing are currently complemented with discriminative rerankers e.g., [1, 2]. The n-best parses generated by the parser for any input sentence (together with their probabilities) are reranked on the basis of rich feature sets and conditional probability estimates. The generative parser, essentialy a joint probability over sentence-parse pairs defined by a generative grammar, are trained on treebanks like the Penn Wall Street Journal (WSJ) [3]. Usually, a corpus consists of language use concerning a range of topics. As observed by [4], subdomains like "politics, stock market, financial news etc. can be found" in the WSJ. When a joint probability (over sentence-parse pairs) is trained on this treebank, the statistics gathered are averages over the different subdomains. By definition, averages smooth-out the statistical differences between the individual subdomains and could possibly make the generative model's task of initial ranking harder.

The present paper explores the question whether there is any gain to be had from incorporating subdomain statistics in parse-reranking. It describes a new method for incorporating subdomain statistics into an existing state-of-the-art parser [1] in order to improve its n-best lists. The main idea is to exploit unannotated, subdomain specific corpora gathered from the web, for weighting the original treebank trees so they reflect subdomain statistics, and employ the resulting weighted treebanks for training individual subdomain sensitive parsers. Our weighting method can be seen as an instance of "Instance Weighting", an idea that surfaced in the context of adaptation [5], but has not been instantiated or tested before (within statistical parsing). In this paper we depart from a formidable parser (Charniak's) and exhibit how it may benefit from "subdomain instance-weighting" for composing its n-best lists.

In what follows we first disucss related work, then we describe the rationale behind instance weighting and define our weighting approach. Subsequently we outline our experimental setting and exihibit results with parsing-reranking the WSJ using the instance weighting technique. Finally we discuss some conclusion from this work.

2. Related Work

An early related study is [10]. Sekine analyzes the "domain dependence of parsing". In his experiments, a domain is characterized by the natural domains defined in the Brown corpus, for example 'Press Reportage', 'General Fiction' or 'Romance and Love Story'. Sekine observes that in parsing, the data from the same domain is the most advantageous, followed by data from the same class, while training on data from another domain generally performs worst. Sekine concludes that when trying "to parse a text in a particular domain, we should prepare a grammar which suits this domain" [10], thus suggesting a "domain-dependent parser".

Although different in flavour, work on domain adaptation is rather related to our work. While domain adaptation aims at adapting a parser from one domain to another, we aim here at finding the influence of specific unlabeled subdomain data on the performance of a "broad-coverage" parser. The role of subdomains and domains in a statistical classifier/model is not exactly the same: in the present work we try to produce specialized subdomain parsers in order to improve parser quality, as opposed to using raw data to migrate the statistics from domain to another.

Recent research on adaptation is too numerous to discuss in detail in this paper. In particular, [5] suggest "instance weighting" as a method for adaptation. They examine their approach on three Natural Language Processing tasks: POS tagging, entity type classification and spam filtering. Our approach, subdomain instance weighting using raw data, can be seen as a novel version thereof for statistical parsing.

Theoretically speaking, successfull domain adaptation hinges on some sense of "overlap" between the source and target domains, e.g., [6]. The overlap between source and target domains can be seen as a (mix of) subdomain(s) of both. Naturally, instance weighting, and its subdomain instantiation, can be seen as a weighted versions of limited self-training, e.g., [2], which is again related to co-training [7, 8].

3. Data and Tools

All experiments were performed using the first-stage generative parser of Charniak [1]. We use the Penn Treebank (PT) Wall Street Journal (WSJ) [3], with the by now 'standard division' into training (sections 02-21) and development/dev (sec-

tion 00). *The current reported results are reported only on the dev set (section 00).* We keep the test set (section 23) for future experiments with more advanced versions. Charniak's parser needs, beside the training set, a heldout set for tuning its pruning parameters, we use here section 24 for that purpose.

Subdomains: As concepts constituting possible subdomains within the PT WSJ we assumed: FINANCIAL, POLITICS and SPORTS. For the POLITICS subdomain we use the English part of the *Europarl Parallel Corpus*[1] [9]. For the FINANCIAL and SPORTS subdomains, to the best of our knowledge there were no ready-to-use corpora available. Hence, we used Wikipedia [11] to create domain-specific corpora ourselves. We used Wikipedia's category system, as provided, to extract relevant articles from the English Wikipedia's dump file, cleaned the articles from Wiki-syntax and segmented them into a one sentence-per-line corpus. The size of the resulting raw domain-specific corpora ranges from 6 to 11 million tokens.

Language Models: For each of the possible subdomains, statistical language models (LMs) were estimated and smoothed (using Chen and Goodman's modified Kneser-Ney smoothing) by using the SRI Language Modeling Toolkit[2] (SRILM) [12]. Using the instance weighting formula 4 we created subdomain specific training data for the subdomain-dependent parsers. The size of the resulting treebanks is between 127k and 160k training instances.

Figure 1 shows our experimental setting. The subdomain-weighted versions of the training treebank are created to train the parsing model. Our ensemble of parsers consists of a total of four parsers: three subdomain-dependent parsers, and the baseline parser. The subdomain-dependent parsers represent the FINANCIAL, POLITICS and SPORTS domains, respectively. The baseline parser is trained on the original treebank, the usual Penn Treebank WSJ sections 02 to 21, and is included in the ensemble to represent the 'general' domain "WSJ rest".

To evaluate performance we use EVALB and the standard PARSEVAL evaluation metrics. Results of parsing sentences of length up to 40 and 100 words are reported for the development (section 00).

4. Subdomain Instance Weighting

Suppose we know of a subdomain $\mathbf{d}$ of the WSJ domain $\mathbf{w}$, then we would like to scale the counts of parses in the WSJ such that those that are more similar to parses found in $\mathbf{d}$ get higher counts than other WSJ parses. Had we had access to a subdomain $\mathbf{d}$ treebank, we could train parser parameters π by maximum-likelihood training:

$$\arg\max_{\pi} \sum_{\langle s,t\rangle \in \mathbf{d}} -P_{\mathbf{d}}(s,t) \log P(s,t;\pi) \tag{1}$$

Where $\langle s,t\rangle$ are sentence-parse pairs. Since we do not have subdomain treebanks (i.e., $P_{\mathbf{d}}(s,t)$), we cannot train the parameters π on complete data from $\mathbf{d}$. While we cannot make assumptions regarding joint sentence-parse probabilities or marginal sentence probabilities, we might get away with the assumption that given a sentence s, conditional parse probabilities $P(t|s)$ do not change much from one domain to another, i.e., $P_{\mathbf{d}}(t|s) \approx P_{\mathbf{w}}(t|s)$. This is equivalent to writing: $\frac{P_{\mathbf{d}}(t,s)}{P_{\mathbf{w}}(t,s)} \approx \frac{P_{\mathbf{d}}(s)}{P_{\mathbf{w}}(s)}$. We can rewrite formula 1:

$$\arg\max_{\pi} \sum_{\langle s,t\rangle \in \mathbf{d}} \frac{P_{\mathbf{d}}(s)}{P_{\mathbf{w}}(s)} P_{\mathbf{w}}(s,t) \log P(s,t;\pi) \tag{2}$$

$$\approx \arg\max_{\pi} \sum_{\langle s,t\rangle \in \mathbf{w}} \frac{P_{\mathbf{d}}(s)}{P_{\mathbf{w}}(s)} P_{\mathbf{w}}(s,t) \log P(s,t;\pi) \tag{3}$$

Hence, we could scale the WSJ parses $\langle s,t\rangle$ by a ratio $\frac{P_{\mathbf{d}}(s)}{P_{\mathbf{w}}(s)}$. The above reasoning follows the line of reasoning leading to "instance-weighting" in [5].

To obtain weights $\frac{P_{\mathbf{d}}(s)}{P_{\mathbf{w}}(s)}$ we will make use of estimates of the two component probabilities $P_x(s)$ for any subdomain x using ngram language models trained over subdomain data. For this purpose we collect raw corpora of the desired subdomains and use these for obtaining language models.

The weights $\frac{P_{\mathbf{d}}(s)}{P_{\mathbf{w}}(s)}$ may often be large numbers which we cannot use for scaling the WSJ parse counts without resulting in data storage problems[3]. Define $D(s) \doteq \log P_{\mathbf{w}}(s) - \log P_{\mathbf{d}}(s)$, for all $\langle s,t\rangle \in \mathbf{w}$ we scale parse counts in the treebank $\mathbf{w}$ by multiplying them with $C_{\mathbf{d}}(s,t)$:

$$C_{\mathbf{d}}(s,t) = \begin{cases} \alpha \times D(s) + \beta & D(s) > 0 \\ 1 & otherwise \end{cases} \tag{4}$$

Where $\alpha = 5$ and $\beta = 500$ set empirically on the dev set data. For smoothing purposes, $C_{\mathbf{d}}(s,t)$ is such that it excludes no trees from $\mathbf{w}$. Under this scheme, the subdomain treebanks contain: SPORTS = 145,474, FINANCIAL =190,376 and POLITICS =177,741 trees.

4.1. Subdomain parsers:

For every subdomain instance weighted treebank (FINANCIAL, POLITICS and SPORTS), we retrain Charniak's parser on that treebank *not* using any heldout set for parser tuning. As expected, the results of each of the subdomain parsers (table 1) are far less accurate than the original WSJ parser because (we hope that) the subdomain parser have specialized in sentences that are more similar to the specific subdomain. Despite these low F-scores[4] we show next that the output of these subdomain parsers is complementary to the more general WSJ parser's output.

Parser	**length$\leq$ 40**	**length$\leq$ 100**
WSJ	90.82	89.87
POLITICS	84.75	82.19
FINANCIAL	84.98	82.73
SPORTS	85.30	83.22

Table 1: F-scores for subdomain parsers on WSJ dev set

[1] Europarl might be suboptimal as the language use of debate transcripts might intuitively differ from the use in journalistic text.

[2] `http://www.speech.sri.com/projects/srilm/`

[3] The parser training programs read-in a sequence of parses rather than parse-count pairs. This means that we are forced to scale our counts into integers, blowing up the treebank counts. We expect this to lead to suboptimal results, also because we loose on the parser-internal smoothing.

[4] The harmonic mean of labeled precision (LP) and labeled recall (LR): F-score $= \frac{2*LP*LR}{LP+LR}$.

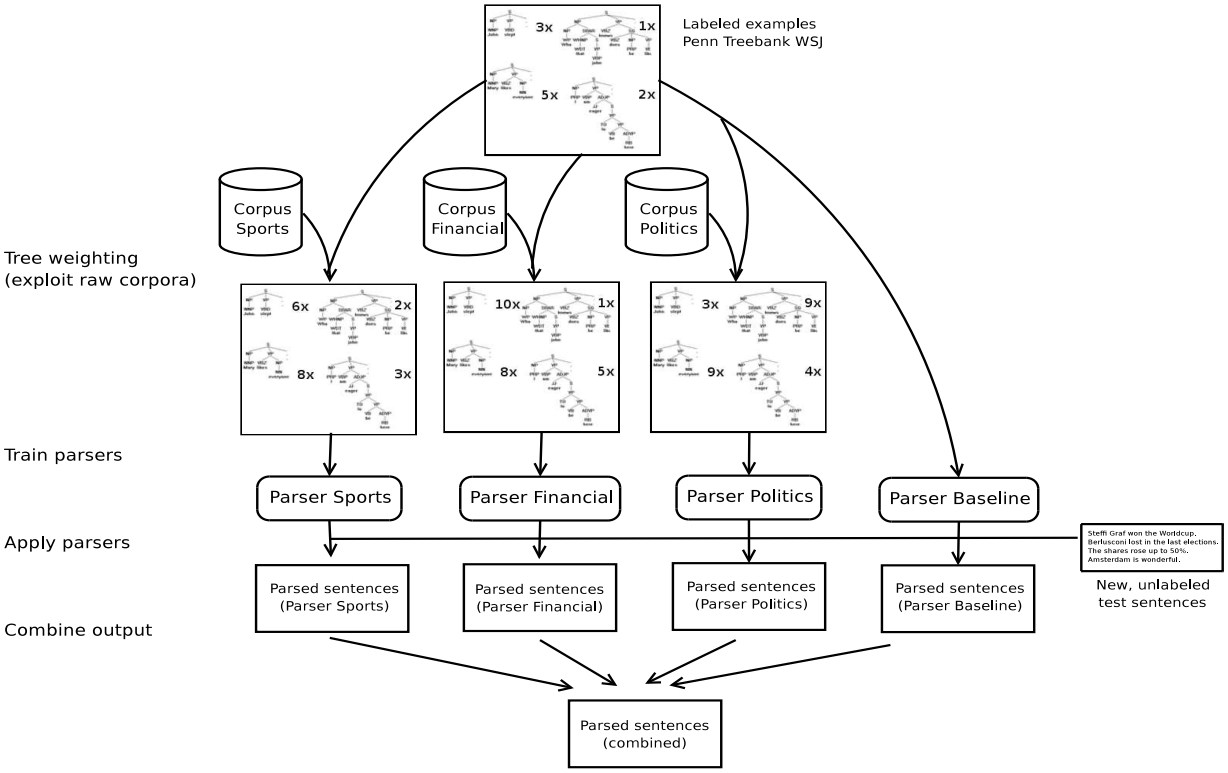

Figure 1: Summary of the Experimental Design for Parsing with Subdomain Instance Weighting

4.2. Union of n-best lists

We gauge in how far the instance weighting method results in subdomains parsers with complementary capabilities, i.e. variance in their n-best lists. We combine the n-best lists of the parsers by taking the union set and measure improvement using an oracle. Given a set of candidate parse trees, the oracle is a decision procedure that selects the best tree by measuring the accuracy in F-score against the gold standard tree.

Figure 2 exibits the results of the oracle both for WSJ parser n-best against the union set of subdomain parsers' output with WSJ n-best. Table 2 shows some of the f-scores. It can be seen, for example, that when the 10-best of the subdomain parsers is united with WSJ 10-best, it is of quality approaching 50-best of the WSJ parser. The 50-best union set of parses (%97.80) is %0.45 higher than 50-best WSJ alone (%97.35) a reduction of about %17 in error at a very high performance level.

n	1	4	10	20	30	40	50
	F-score length ≤ 40						
WSJ	90.82	94.33	95.86	96.60	97.00	97.20	97.35
Union	92.83	95.45	96.74	97.30	97.54	97.68	97.80
	F-score length ≤ 100						
WSJ	89.87	93.29	94.91	95.70	96.09	96.32	96.50
Union	91.78	94.43	95.77	96.36	96.61	96.79	96.94

Table 2: Oracle F-scores for n-best WSJ parser vs. Union output of four subdomain parsers.

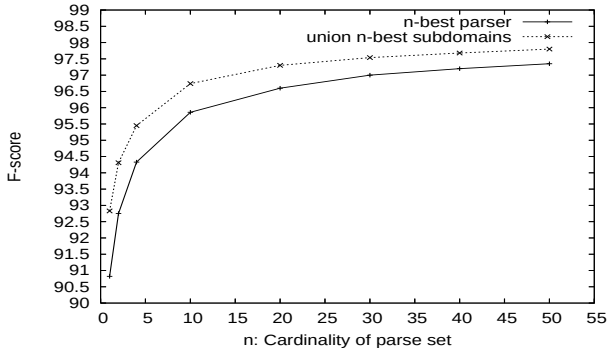

Figure 2: Oracle F-score for WSJ vs. union set of subdomain parsers as function of n-best

5. Is subdomain probability a discriminative feature?

The fact that the union of the n-best lists of the subdomain parsers gives improved oracle results is encouraging given the simplicity of instance weighting. Nothing prevents us from applying reranking for selecting a parse from the union set of parses. It seems to us more expedient if the reranker has access also to new discriminative features of the subdomain data. However, this constitutes a research agenda beyond the scope of this short paper (improved generative n-best parsing).

5.1. Parse probability for n-best:

We select from the union set of parses (output by all subdomain parsers) exactly the n parses (for the n values in Table 2) with the highest probabilities given by any of the subdomain parsers. With this simple selection procedure we obtain up to 0.04 improvement over Charniak's n-best, a meager yet meaningful result given the simplicity of the selection procedure: the parse probabilities output by the diverse parsers (WSJ and the subdomain parsers) might be useful as a feature for a discriminative reranker.

5.2. Adding a single parse to n-best (n+1-best):

In this experiment we simply added to the WSJ parser's n-best the single most probable parse output by the FINANCIAL subdomain parser. The gain in F-score between the (n+1) parses and the n-best is given in Table 3. Adding the most probable parse from the other two subdomains gives a smaller gain, an expected outcome given that FINANCIAL is closest to WSJ, when compared to SPORTS and POLITICS (EuroParl). As expected, most gain comes at a lower values of n, and as n grows, the gain diminishes.

n	1	4	10	20	30	40	50
Gain	+1.05	+0.29	+0.13	+0.11	+0.08	+0.08	+0.03

Table 3: F-score gain by adding the single most probable parse from the FINANCIAL parser to n-best.

Clearly, the probability given by the subdomain parser has reasonable discriminative power, which is encouraging given the suboptimal weighting formula[5] (formula 4 as opposed to the theoretical $\frac{P_{\mathbf{d}}(s)}{P_{\mathbf{w}}(s)}$ from section 4).

6. Conclusions and Outlook

This paper explores a particular instantiation for subdomain instance weighting for n-best parsing. Our approach exploits unlabeled subdomain corpora for obtaining statistics for applying instance weighting to a treebank. The empirical results warrant the conclusion that the idea of subdomain instance weighting is worthwhile further exploration. The experiments show that a generative statistical parser could benefit from raw subdomain data in its model.

Future work will explore other ways of instantiating subdomain instance weighting for parsing (e.g. latent subdomains), by extending the current approach to parse-reranking, and examining how the current approach can be used for domain adaptation.

7. References

[1] Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of ACL 2005*. The Association for Computer Linguistics.

[2] D. McClosky, E. Charniak, and M. Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of the COLING-ACL 2006*. The Association for Computer Linguistics.

[3] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

[4] R. Kneser and J. Peters. 1997. Semantic clustering for adaptive language modeling. In *ICASSP 1997*, volume 02, page 779, Los Alamitos, CA, USA. IEEE Computer Society.

[5] Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of ACL 2007*, pages 264–271, Prague, Czech Republic, June. Association for Computational Linguistics.

[6] Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

[7] Avrim Blum and Tom M. Mitchell. 1998. Combining labeled and unlabeled sata with co-training. In *COLT*, pages 92–100.

[8] Abney, S. (2007). Semisupervised Learning for Computational Linguistics. Chapman & Hall/CRC Computer Science & Data Analysis Series 8. CRC Press.

[9] Philipp Koehn. 2005. Europarl: A Multilingual Corpus for Evaluation of Machine Translation. In *Proceedings MT Summit 2005*.

[10] Sekine, S. (1997). The Domain Dependence of Parsing. Washington, DC, USA. The Fifth Conference on Applied Natural Language Processing.

[11] Wikimedia Foundation Inc. 2007. Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/.

[12] Stolcke, A. (2002) SRILM - An Extensible Language Modeling Toolkit. Denver, Colorado. Proc. Intl. Conf. on Spoken Language Processing, vol. 2, pp. 901-904.

[5] Chosen largely to suit the training program of the parser which does not read in parse-count pairs.