An Ensemble Method for Selection of High Quality Parses

Roi Reichart
ICNC
Hebrew University of Jerusalem
roiri@cs.huji.ac.il

Ari Rappoport
Institute of Computer Science
Hebrew University of Jerusalem
arir@cs.huji.ac.il

Abstract

While the average performance of statistical parsers gradually improves, they still attach to many sentences annotations of rather low quality. The number of such sentences grows when the training and test data are taken from different domains, which is the case for major web applications such as information retrieval and question answering.

In this paper we present a *Sample Ensemble Parse Assessment (SEPA)* algorithm for detecting parse quality. We use a function of the agreement among several copies of a parser, each of which trained on a different sample from the training data, to assess parse quality. We experimented with both generative and reranking parsers (Collins, Charniak and Johnson respectively). We show superior results over several baselines, both when the training and test data are from the same domain and when they are from different domains. For a test setting used by previous work, we show an error reduction of 31% as opposed to their 20%.

1 Introduction

Many algorithms for major NLP applications such as information extraction (IE) and question answering (QA) utilize the output of statistical parsers (see (Yates et al., 2006)). While the average performance of statistical parsers gradually improves, the quality of many of the parses they produce is too low for applications. When the training and test data are taken from different domains (the *parser adaptation* scenario) the ratio of such low quality parses becomes even higher. Figure 1 demonstrates these phenomena for two leading models, Collins (1999) model 2, a generative model, and Charniak and Johnson (2005), a reranking model. The parser adaptation scenario is the rule rather than the exception for QA and IE systems, because these usually operate over the highly variable Web, making it very difficult to create a representative corpus for manual annotation. Medium quality parses may seriously harm the performance of such systems.

In this paper we address the problem of assessing parse quality, using a *Sample Ensemble Parse Assessment (SEPA)* algorithm. We use the level of agreement among several copies of a parser, each of which trained on a different sample from the training data, to predict the quality of a parse. The algorithm does not assume uniformity of training and test data, and is thus suitable to web-based applications such as QA and IE.

Generative statistical parsers compute a probability $p(a, s)$ for each sentence annotation, so the immediate technique that comes to mind for assessing parse quality is to simply use $p(a, s)$. Another seemingly trivial method is to assume that shorter sentences would be parsed better than longer ones. However, these techniques produce results that are far from optimal. In Section 5 we show the superiority of our method over these and other baselines.

Surprisingly, as far as we know there is only one previous work explicitly addressing this problem (Yates et al., 2006). Their WOODWARD algorithm filters out high quality parses by performing seman-

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 408–415,
Prague, Czech Republic, June 2007.

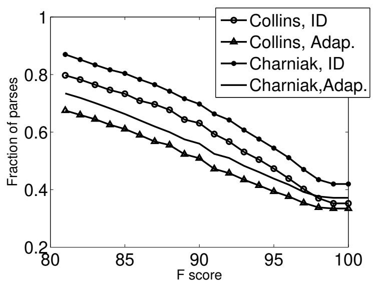

Figure 1: F-score vs. the fraction of parses whose f-score is at least that f-score. For the in-domain scenario, the parsers are tested on sec 23 of the WSJ Penn Treebank. For the parser adaptation scenario, they are tested on the Brown test section. In both cases they are trained on sections 2-21 of WSJ.

tic analysis. The present paper provides a detailed comparison between the two algorithms, showing both that SEPA produces superior results and that it operates under less restrictive conditions.

We experiment with both the generative parsing model number 2 of Collins (1999) and the reranking parser of Charniak and Johnson (2005), both when the training and test data belong to the same domain (the *in-domain* scenario) and in the parser adaptation scenario. In all four cases, we show substantial improvement over the baselines. The present paper is the first to use a reranking parser and the first to address the adaptation scenario for this problem.

Section 2 discusses relevant previous work, Section 3 describes the SEPA algorithm, Sections 4 and 5 present the experimental setup and results, and Section 6 discusses certain aspects of these results and compares SEPA to WOODWARD.

2 Related Work

The only previous work we are aware of that explicitly addressed the problem of detecting high quality parses in the output of statistical parsers is (Yates et al., 2006). Based on the observation that incorrect parses often result in implausible semantic interpretations of sentences, they designed the WOODWARD filtering system. It first maps the parse produced by the parser to a logic-based representation (relational conjunction (RC)) and then employs four methods for semantically analyzing whether a conjunct in the RC is likely to be reasonable. The filters use semantic information obtained from the Web. Measuring errors using filter f-score (see Section 3) and using the Collins generative model, WOODWARD reduces errors by 67% on a set of TREC questions and by 20% on a set of a 100 WSJ sentences. Section 5 provides a detailed comparison with our algorithm.

Reranking algorithms (Koo and Collins, 2005; Charniak and Johnson, 2005) search the list of best parses output by a generative parser to find a parse of higher quality than the parse selected by the generative parser. Thus, these algorithms in effect assess parse quality using syntactic and lexical features. The SEPA algorithm does not use such features, and is successful in detecting high quality parses even when working on the output of a reranker. Reranking and SEPA are thus relatively independent.

Bagging (Breiman, 1996) uses an ensemble of instances of a model, each trained on a sample of the training data[1]. Bagging was suggested in order to enhance classifiers; the classification outcome was determined using a majority vote among the models. In NLP, bagging was used for active learning for text classification (Argamon-Engelson and Dagan, 1999; McCallum and Nigam, 1998). Specifically in parsing, (Henderson and Brill, 2000) applied a constituent level voting scheme to an ensemble of bagged models to increase parser performance, and (Becker and Osborne, 2005) suggested an active learning technique in which the agreement among an ensemble of bagged parsers is used to predict examples valuable for human annotation. They reported experiments with small training sets only (up to 5,000 sentences), and their agreement function is very different from ours. Both works experimented with generative parsing models only.

Ngai and Yarowsky (2000) used an ensemble based on bagging and partitioning for active learning for base NP chunking. They select top items without any graded assessment, and their f-complement function, which slightly resembles our MF (see the next section), is applied to the output of a classifier, while our function is applied to structured output. A survey of several papers dealing with mapping

[1] Each sample is created by sampling, with replacement, L examples from the training pool, where L is the size of the training pool. Conversely, each of our samples is smaller than the training set, and is created by sampling without replacement. See Section 3 ('regarding S') for a discussion of this issue.

predictors in classifiers' output to posterior probabilities is given in (Caruana and Niculescu-Mizil, 2006). As far as we know, the application of a sample based parser ensemble for assessing parse quality is novel.

Many IE and QA systems rely on the output of parsers (Kwok et al., 2001; Attardi et al., 2001; Moldovan et al., 2003). The latter tries to address incorrect parses using complex relaxation methods. Knowing the quality of a parse could greatly improve the performance of such systems.

3 The Sample Ensemble Parse Assessment (SEPA) Algorithm

In this section we detail our parse assessment algorithm. Its input consists of a parsing algorithm A, an annotated training set TR, and an unannotated test set TE. The output provides, for each test sentence, the parse generated for it by A when trained on the full training set, and a grade assessing the parse's quality, on a continuous scale between 0 to 100. Applications are then free to select a sentence subset that suits their needs using our grades, e.g. by keeping only high-quality parses, or by removing low-quality parses and keeping the rest. The algorithm has the following stages:

1. Choose N random samples of size S from the training set TR. Each sample is selected without replacement.
2. Train N copies of the parsing algorithm A, each with one of the samples.
3. Parse the test set with each of the N models.
4. For each test sentence, compute the value of an agreement function F between the models.
5. Sort the test set according to F's value.

The algorithm uses the level of agreement among several copies of a parser, each trained on a different sample from the training data, to predict the quality of a parse. The higher the agreement, the higher the quality of the parse. Our approach assumes that if the parameters of the model are well designed to annotate a sentence with a high quality parse, then it is likely that the model will output the same (or a highly similar) parse even if the training data is somewhat changed. In other words, we rely on the stability of the parameters of statistical parsers. Although this is not always the case, our results confirm that strong correlation between agreement and parse quality does exist.

We explored several agreement functions. The one that showed the best results is *Mean F-score (MF)*[2], defined as follows. Denote the models by $m_1 \dots m_N$, and the parse provided by m_i for sentence s as $m_i(s)$. We randomly choose a model m_l, and compute

$$MF(s) = \frac{1}{N-1} \sum_{i \in [1 \dots N], i \neq l} fscore(m_i, m_l) \quad (1)$$

We use two measures to evaluate the quality of SEPA grades. Both measures are defined using a threshold parameter T, addressing only sentences whose SEPA grades are not smaller than T. We refer to these sentences as *T-sentences*.

The first measure is the average f-score of the parses of T-sentences. Note that we compute the f-score of each of the selected sentences and then average the results. This stands in contrast to the way f-score is ordinarily calculated, by computing the labeled precision and recall of the constituents in the whole set and using these as the arguments of the f-score equation. The ordinary f-score is computed that way mostly in order to overcome the fact that sentences differ in length. However, for applications such as IE and QA, which work at the single sentence level and which might reach erroneous decision due to an inaccurate parse, normalizing over sentence lengths is less of a factor. For this reason, in this paper we present detailed graphs for the average f-score. For completeness, Table 4 also provides some of the results using the ordinary f-score.

The second measure is a generalization of the filter f-score measure suggested by Yates et al. (2006). They define *filter precision* as the ratio of correctly parsed sentences in the *filtered set* (the set the algorithm choose) to total sentences in the filtered set and *filter recall* as the ratio of correctly parsed sentences in the filtered set to correctly parsed sentences in the

[2] Recall that sentence f-score is defined as: $f = \frac{2 \times P \times R}{P+R}$, where P and R are the labeled precision and recall of the constituents in the sentence relative to another parse.

whole set of sentences parsed by the parser (*unfiltered set* or *test set*). Correctly parsed sentences are sentences whose parse got f-score of 100%.

Since requiring a 100% may be too restrictive, we generalize this measure to *filter f-score with parameter k*. In our measure, the filter recall and precision are calculated with regard to sentences that get an f-score of k or more, rather than to correctly parsed sentences. Filtered f-score is thus a special case of our filtered f-score, with parameter 100.

We now discuss the effect of the number of models N and the sample size S. The discussion is based on experiments (using development data, see Section 4) in which all the parameters are fixed except for the parameter in question, using our development sections.

Regarding N (see Figure 2): As the number of models increases, the number of T-sentences selected by SEPA decreases and their quality improves, in terms of both average f-score and filter f-score (with $k = 100$). The fact that more models trained on different samples of the training data agree on the syntactic annotation of a sentence implies that this syntactic pattern is less sensitive to perturbations in the training data. The number of such sentences is small and it is likely the parser will correctly annotate them. The smaller T-set size leads to a decrease in filter recall, while the better quality leads to an increase in filter precision. Since the increase in filter precision is sharper than the decrease in filter recall, filter f-score increases with the number of models N.

Regarding S^3: As the sample size increases, the number of T-sentences increases, and their quality degrades in terms of average f-score but improves in terms of filter f-score (again, with parameter $k = 100$). The overlap among smaller samples is small and the data they supply is sparse. If several models trained on such samples attach to a sentence the same parse, this syntactic pattern must be very prominent in the training data. The number of such sentences is small and it is likely that the parser will correctly annotate them. Therefore smaller sample size leads to smaller T-sets with high average f-score. As the sample size increases, the T-set becomes larger but the average f-score of a parse decreases. The larger T-set size leads to increase in filter recall, while the lower average quality leads to decrease in filter precision. Since the increase in filter recall is sharper than the decrease in filter precision, the result is that filter f-score increases with the sample size S.

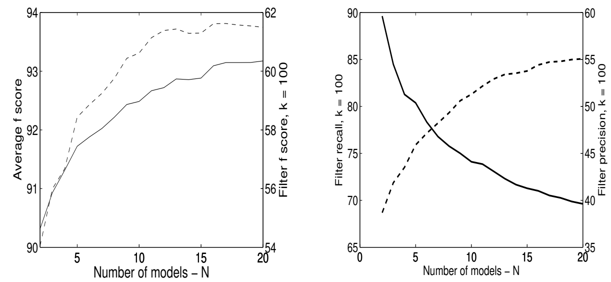

Figure 2: The effect of the number of models N on SEPA (Collins' model). The scenario is in-domain, sample size $S = 33,000$ and $T = 100$. We see: average f-score of T-sentences (left, solid curve and left y-axis), filter f-score with $k = 100$ (left, dashed curve and right y-axis), filter recall with $k = 100$ (right, solid curve and left y-axis), and filter precision with $k = 100$ (right, dashed curve and right y-axis).

This discussion demonstrates the importance of using both average f-score and filter f-score, since the two measures reflect characteristics of the selected sample that are not necessarily highly (or positively) correlated.

4 Experimental Setup

We performed experiments with two parsing models, the Collins (1999) generative model number 2 and the Charniak and Johnson (2005) reranking model. For the first we used a reimplementation (?). We performed experiments with each model in two scenarios, in-domain and parser adaptation. In both experiments the training data are sections 02-21 of the WSJ PennTreebank (about 40K sentences). In the in-domain experiment the test data is section 23 (2416 sentences) of WSJ and in the parser adaptation scenario the test data is Brown test section (2424 sentences). Development sections are WSJ section 00 for the in-domain scenario (1981 sentences) and Brown development section for the adaptation scenario (2424 sentences). Following

[3] Graphs are not shown due to lack of space.

(Gildea, 2001), the Brown test and development sections consist of 10% of Brown sentences (the 9th and 10th of each 10 consecutive sentences in the development and test sections respectively).

We performed experiments with many configurations of the parameters N (number of models), S (sample size) and F (agreement function). Due to space limitations we describe only experiments where the values of the parameters N, S and F are fixed (F is MF, N and S are given in Section 5) and the threshold parameter T is changed.

5 Results

We first explore the quality of the selected set in terms of average f-score. In Section 3 we reported that the quality of a selected T-set of parses increases as the number of models N increases and sample size S decreases. We therefore show the results for relatively high N (20) and relatively low S (13,000, which is about a third of the training set). Denote the cardinality of the set selected by SEPA by n (it is actually a function of T but we omit the T in order to simplify notations).

We use several baseline models. The first, *confidence baseline (CB)*, contains the n sentences having the highest parser assigned probability (when trained on the whole training set). The second, *minimum length (ML)*, contains the n shortest sentences in the test set. Since many times it is easier to parse short sentences, a trivial way to increase the average f-score measure of a set is simply to select short sentences. The third, following (Yates et al., 2006), is *maximum recall (MR)*. MR simply predicts that all test set sentences should be contained in the selected T-set. The output set of this model gets filter recall of 1 for any k value, but its precision is lower. The MR baseline is not relevant to the average f-score measure, because it selects all of the sentences in a set, which leads to the same average as a random selection (see below). In order to minimize visual clutter, for the filter f-score measure we use the maximum recall (MR) baseline rather than the minimum length (ML) baseline, since the former outperforms the latter. Thus, ML is only shown for the average f-score measure. We have also experimented with a random baseline model (containing n randomly selected test sentences), whose results are the worst and which is shown for reference.

Readers of this section may get confused between the agreement threshold parameter T and the parameter k of the filter f-score measure. Please note: as to T, SEPA sorts the test set by the values of the agreement function. One can then select only sentences whose agreement score is at least T. T's values are on a continuous scale from 0 to 100. As to k, the filter f-score measure gives a grade. This grade combines three values: (1) the number of sentences in the set (selected by an algorithm) whose f-score relative to the gold standard parse is at least k, (2) the size of the selected set, and (3) the total number of sentences with such a parse in the whole test set. We did not introduce separate notations for these values.

Figure 3 (top) shows average f-score results where SEPA is applied to Collins' generative model in the in-domain (left) and adaptation (middle) scenarios. SEPA outperforms the baselines for all values of the agreement threshold parameter T. Furthermore, as T increases, not only does the SEPA set quality increase, but the quality differences between this set and the baseline sets increases as well. The graphs on the right show the number of sentences in the sets selected by SEPA for each T value. As expected, this number decreases as T increases.

Figure 3 (bottom) shows the same pattern of results for the Charniak reranking parser in the in-domain (left) and adaptation (middle) scenarios. We see that the effects of the reranker and SEPA are relatively independent. Even after some of the errors of the generative model were corrected by the reranker by selecting parses of higher quality among the 50-best, SEPA can detect parses of high quality from the set of parsed sentences.

To explore the quality of the selected set in terms of filter f-score, we recall that the quality of a selected set of parses increases as both the number of models N and the sample size S increase, and with T. Therefore, for $k = 85 \ldots 100$ we show the value of filter f-score with parameter k when the parameters configuration is a relatively high N (20), relatively high S (33,000, which are about 80% of the training set), and the highest T (100).

Figure 4 (top) shows filter f-score results for Collins' generative model in the in-domain (left) and adaptation (middle) scenarios. As these graphs show, SEPA outperforms CB and random for all values of the filter f-score parameter k, and outperforms the MR baseline where the value of k is 95 or more. Although for small k values MR gets a higher f-score than SEPA, the filter precision of SEPA is much higher (right, shown for adaptation. The in-domain pattern is similar and not shown). This stems from the definition of the MR baseline, which simply predicts any sentence to be in the selected set. Furthermore, since the selected set is meant to be the input for systems that require high quality parses, what matters most is that SEPA outperforms the MR baseline at the high k ranges.

Figure 4 (bottom) shows the same pattern of results for the Charniak reranking parser in the in-domain (left) and adaptation (middle) scenarios. As for the average f-score measure, it demonstrates that the effects of the reranker and SEPA algorithm are relatively independent.

Tables 1 and 2 show the error reduction achieved by SEPA for the filter f-score measure with parameters $k = 95, 97, 100$ (Table 1) and for the average f-score measure with several SEPA agreement threshold (T) values (Table 2) . The error reductions achieved by SEPA for both measures are substantial.

Table 3 compares SEPA and WOODWARD on the exact same test set used by (Yates et al., 2006) (taken from WSJ sec 23). SEPA achieves error reduction of 31% over the MR baseline on this set, compared to only 20% achieved by WOODWARD. Not shown in the table, in terms of ordinary f-score WOODWARD achieves error reduction of 37% while SEPA achieves 43%. These numbers were the only ones reported in (Yates et al., 2006).

For completeness of reference, Table 4 shows the superiority of SEPA over CB in terms of the usual f-score measure used by the parsing community (numbers are counted for constituents first). Results for other baselines are even more impressive. The configuration is similar to that of Figure 3.

6 Discussion

In this paper we introduced SEPA, a novel algorithm for assessing parse quality in the output of a statistical parser. SEPA is the first algorithm shown to be successful when a reranking parser is considered, even though such models use a reranker to detect and fix some of the errors made by the base generative model. WOODWARD, the only previously suggested algorithm for this problem, was tested with Collins' generative model only. Furthermore, this is the first time that an algorithm for this problem succeeds in a domain adaptation scenario, regardless of

	Filter f-score					
	In-domain			Adaptation		
k value	95	97	100	95	97	100
Coll. MR	3.5	20.1	29.2	22.8	29.8	33.6
Coll. CB	11.6	11.7	3.4	14.2	9.9	7.4
Char. MR	1.35	13.6	23.44	21.9	30	32.5
Char. CB	21.9	16.8	11.9	25	20.2	16.2

Table 1: Error reduction in the filter f-score measure obtained by SEPA with Collins' (top two lines) and Charniak's (bottom two lines) model, in the two scenarios (in-domain and adaptation), vs. the maximum recall (MR lines 1 and 3) and confidence (CB, lines 2 and 4) baselines, using $N = 20, T = 100$ and $S = 33,000$. Shown are parameter values $k = 95, 97, 100$. Error reduction numbers were computed by $100 \times (fscoreSEPA - fscorebaseline)/(1 - fscorebaseline)$.

	Average f-score					
	In-domain			Adaptation		
T	95	97	100	95	97	100
Coll. ML	32.6	37.2	60.8	46.8	52.7	70.7
Coll. CB	26.5	31.4	53.9	46.9	53.6	70
Char. ML	25.1	33.2	58.5	46.9	58.4	77.1
Char. CB	20.4	30	52	44.4	55.5	73.5

Table 2: Error reduction in the average f-score measure obtained by SEPA with Collins (top two lines) and Charniak (bottom two lines) model, in the two scenarios (in-domain and adaptation), vs. the minimum length (ML lines 1 and 3) and confidence (CB, lines 2 and 4) baselines, using $N = 20$ and $S = 13,000$. Shown are agreement threhsold parameter values $T = 95, 97, 100$. Error reduction numbers were computed by $100 \times (fscoreSEPA - fscorebaseline)/(1 - fscorebaseline)$.

	SEPA	WOODWARD	CB
ER	**31%**	20%	-31%

Table 3: Error reduction compared to the MR baseline, measured by filter f-score with parameter 100. The data is the WSJ sec 23 test set usd by (Yates et al., 2006). All three methods use Collins' model. SEPA uses $N = 20, S = 33,000, T = 100$.

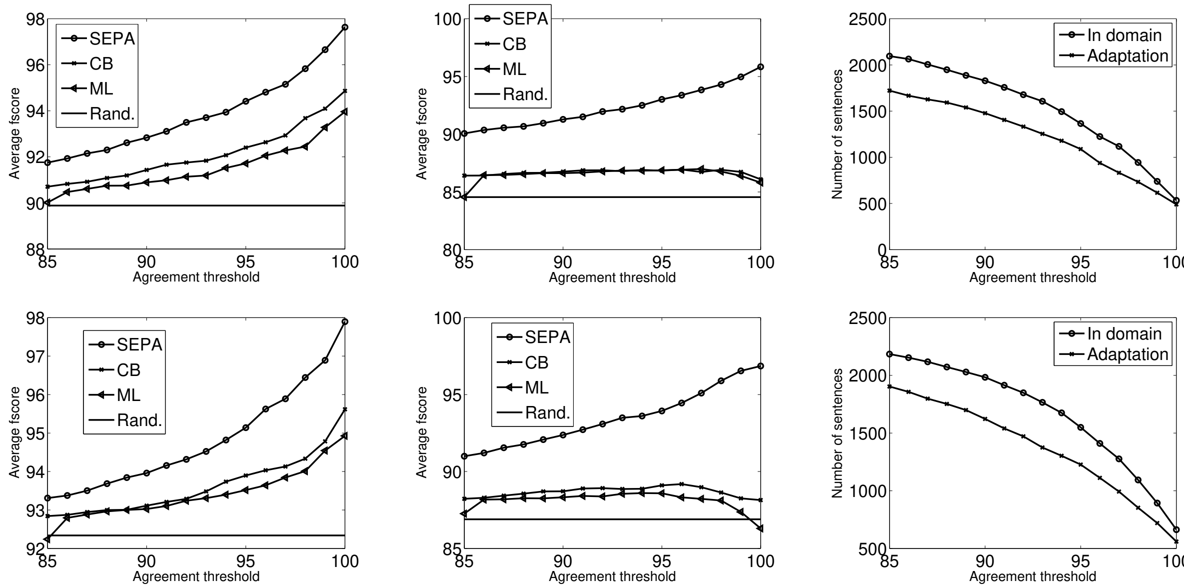

Figure 3: Agreement threshold T vs. average f-score (left and middle) and number of sentences in the selected set (right), for SEPA with Collins' generative model (top) and the Charniak reranking model (bottom). SEPA parameters are $S = 13,000, N = 20$. In both rows, SEPA results for the in-domain (left) and adaptation (middle) scenarios are compared to the confidence (CB) and minimum length (ML) baselines. The graphs on the right show the number of sentences in the selected set for both scenarios.

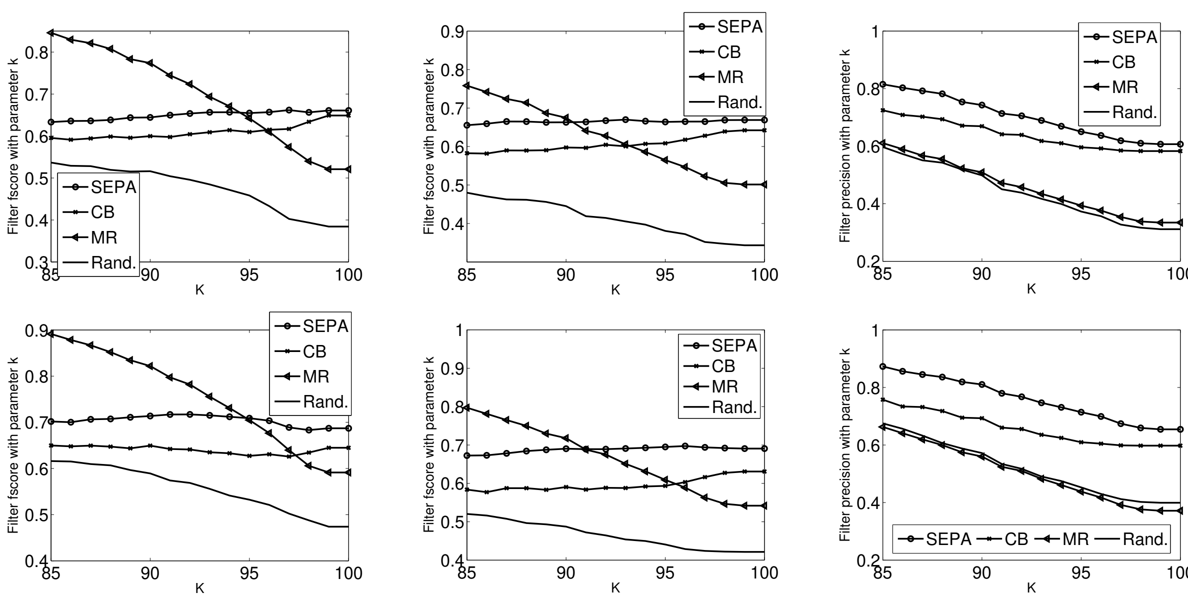

Figure 4: Parameter k vs. filter f-score (left and middle) and filter precision (right) with that parameter, for SEPA with Collins' generative model (top) and the Charniak reranking model (bottom). SEPA parameters are $S = 33,000, N = 20, T = 100$. In both rows, results for the in-domain (left) and adaptation (middle) scenarios. In two leftmost graphs, the performance of the algorithm is compared to the confidence baseline (CB) and maximum recall (MR). The graphs on the right compare the filter precision of SEPA with that of the MR and CB baselines.

the parsing model. In the Web environment this is the common situation.

The WSJ and Brown experiments performed with SEPA are much broader than those performed with WOODWARD, considering all sentences of WSJ sec 23 and Brown test section rather than a subset of carefully selected sentences from WSJ sec 23. However, we did not perform a TREC experiment, as (Yates et al., 2006) did. Our WSJ and Brown results outperformed several baselines. Moreover, WSJ (or Brown) sentences that contain conjunctions were avoided in the experiments of (Yates et al., 2006). We have verified that our algorithm shows substantial error reduction over the baselines for this type of sentences (in the ranges $13 - 46\%$ for the filter f-score with $k = 100$, and $30 - 60\%$ for the average f-score).

As Table 3 shows, on a WSJ sec 23 test set similar to that used by (Yates et al., 2006), SEPA achieves 31% error reduction compared to 20% of WOODWARD.

WOODWARD works under several assumptions. Specifically, it requires a corpus whose content overlaps at least in part with the content of the parsed sentences. This corpus is used to extract semantically related statistics for its filters. Furthermore, the filters of this algorithm (except of the QA filter) are focused on verb and preposition relations. Thus, it is more natural for it to deal with mistakes contained in such relations. This is reflected in the WSJ based test set on which it is tested. SEPA does not make any of these assumptions. It does not use any external information source and is shown to select high quality parses from diverse sets.

	In-domain		Adaptation	
	F	ER	F	ER
SEPA Collins	97.09	44.36%	95.38	66.38%
CB Collins	94.77	–	86.3	–
SEPA Charniak	97.21	35.69%	96.3	54.66%
CB Charniak	95.6	–	91.84	–

Table 4: SEPA error reduction vs. the CB baseline in the in-domain and adaptation scenarios, using the traditional f-score of the parsing literature. $N = 20, S = 13,000, T = 100$.

For future work, integrating SEPA into the reranking process seems a promising direction for enhancing overall parser performance.

Acknowledgement. We would like to thank Dan Roth for his constructive comments on this paper.

References

Shlomo Argamon-Engelson and Ido Dagan, 1996. committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research*, 11:335–360.

Giuseppe Attardi, Antonio Cisternino, Francesco Formica, Maria Simi and Alessandro Tommasi, 2001. PiQASso: Pisa question answering system. *TREC '01*.

Markus Becker and Miles Osborne, 2005. A two-stage method for active learning of statistical grammars. *IJCAI '05*.

Daniel Bikel, 2004. *Code developed at University of Pennsylvania*. http://www.cis.upenn.edu.bikel.

Leo Breiman, 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

Rich Caruana and Alexandru Niculescu-Mizil, 2006. An empirical comparison of supervised learning algorithms. *ICML '06*.

Eugene Charniak and Mark Johnson, 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. *ACL '05*.

Michael Collins, 1999. *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania.

Daniel Gildea, 2001. Corpus variation and parser performance. *EMNLP '01*.

John C. Henderson and Eric Brill, 2000. Bagging and boosting a treebank parser. *NAACL '00*.

Terry Koo and Michael Collins, 2005. Hidden-variable models for discriminative reranking. *EMNLP '05*.

Cody Kwok, Oren Etzioni and Daniel S. Weld, 2001. Scaling question answering to the web. *WWW '01*.

Andrew McCallum and Kamal Nigam, 1998. Employing EM and pool-based active learning for text classification. *ICML '98*.

Dan Moldovan, Christine Clark, Sanda Harabagiu and Steve Maiorano, 2003. Cogex: A logic prover for question answering. *HLT-NAACL '03*.

Grace Ngai and David Yarowsky, 2000. Rule writing or annotation: cost-efficient resource usage for base noun phrase chunking. *ACL '00*.

Alexander Yates, Stefan Schoenmackers and Oren Etzioni, 2006. Detecting parser errors using web-based semantic filters. *EMNLP '06*.