Sample Selection for Statistical Parsers: Cognitively Driven Algorithms and Evaluation Measures

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

Creating large amounts of manually annotated training data for statistical parsers imposes heavy cognitive load on the human annotator and is thus costly and error prone. It is hence of high importance to decrease the human efforts involved in creating training data without harming parser performance. For constituency parsers, these efforts are traditionally evaluated using the total number of constituents (TC) measure, assuming uniform cost for each annotated item. In this paper, we introduce novel measures that quantify aspects of the cognitive efforts of the human annotator that are not reflected by the TC measure, and show that they are well established in the psycholinguistic literature. We present a novel *parameter based sample selection* approach for creating good samples in terms of these measures. We describe methods for global optimisation of lexical parameters of the sample based on a novel optimisation problem, the *constrained multiset multicover* problem, and for *cluster-based sampling* according to syntactic parameters. Our methods outperform previously suggested methods in terms of the new measures, while maintaining similar TC performance.

1 Introduction

State of the art statistical parsers require large amounts of manually annotated data to achieve good performance. Creating such data imposes heavy cognitive load on the human annotator and is thus costly and error prone. Statistical parsers are major components in NLP applications such as QA (Kwok et al., 2001), MT (Marcu et al., 2006) and SRL (Toutanova et al., 2005). These often operate over the highly variable Web, which consists of texts written in many languages and genres. Since the performance of parsers markedly degrades when training and test data come from different domains (Lease and Charniak, 2005), large amounts of training data from each domain are required for using them effectively. Thus, decreasing the human efforts involved in creating training data for parsers without harming their performance is of high importance.

In this paper we address this problem through *sample selection*: given a parsing algorithm and a large pool of unannotated sentences S, select a subset $S_1 \subset S$ for human annotation such that the human efforts in annotating S_1 are minimized while the parser performance when trained with this sample is maximized.

Previous works addressing training sample size vs. parser performance for constituency parsers (Section 2) evaluated training sample size using the total number of constituents (TC). Sentences differ in length and therefore in annotation efforts, and it has been argued (see, e.g, (Hwa, 2004)) that TC reflects the number of decisions the human annotator makes when syntactically annotating the sample, assuming uniform cost for each decision.

In this paper we posit that important aspects of the efforts involved in annotating a sample are not reflected by the TC measure. Since annotators analyze sentences rather than a bag of constituents, sentence structure has a major impact on their cognitive efforts. Sizeable psycholinguistic literature points to the connection between nested structures in the syntactic structure of a sentence and its annotation efforts. This has motivated us to introduce (Section 3) three sample size measures, the total and av-

Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL), pages 3–11,
Boulder, Colorado, June 2009.

erage number of nested structures of degree k in the sample, and the average number of constituents per sentence in the sample.

Active learning algorithms for sample selection focus on sentences that are difficult for the parsing algorithm when trained with the available training data (Section 2). In Section 5 we show that active learning samples contain a high number of complex structures, much higher than their number in a randomly selected sample that achieves the same parser performance level. To avoid that, we introduce (Section 4) a novel *parameter based sample selection* (PBS) approach which aims to select a sample that enables good estimation of the model parameters, without focusing on difficult sentences. In Section 5 we show that the methods derived from our approach select substantially fewer complex structures than active learning methods and the random baseline.

We propose two different methods. In *cluster based sampling* (CBS), we aim to select a sample in which the distribution of the model parameters is similar to their distribution in the whole unlabelled pool. To do that we build a vector representation for each sentence in the unlabelled pool reflecting the distribution of the model parameters in this sentence, and use a clustering algorithm to divide these vectors into clusters. In the second method we use the fact that a sample containing many examples of a certain parameter yields better estimation of this parameter. If this parameter is crucial for model performance and the selection process does not harm the distribution of other parameters, then the selected sample is of high quality. To select such a sample we introduce a reduction between this selection problem and a variant of the NP-hard multiset-multicover problem (Hochbaum, 1997). We call this problem the *constrained multiset multicover* (CMM) problem, and present an algorithm to approximate it.

We experiment (Section 5) with the WSJ PennTreebank (Marcus et al., 1994) and Collins' generative parser (Collins, 1999), as in previous work. We show that PBS algorithms achieve good results in terms of both the traditional TC measure (significantly better than the random selection baseline and similar to the results of the state of the art tree entropy (TE) method of (Hwa, 2004)) and our novel cognitively driven measures (where PBS algorithms significantly outperform both TE and the random baseline). We thus argue that PBS provides a way to select a sample that imposes reduced cognitive load on the human annotator.

2 Related Work

Previous work on sample selection for statistical parsers applied active learning (AL) (Cohn and Ladner, 1994) to corpora of various languages and syntactic annotation schemes and to parsers of different performance levels. In order to be able to compare our results to previous work targeting high parser performance, we selected the corpus and parser used by the method reporting the best results (Hwa, 2004), WSJ and Collins' parser.

Hwa (2004) used uncertainty sampling with the tree entropy (TE) selection function[1] to select training samples for the Collins parser. In each iteration, each of the unlabelled pool sentences is parsed by the parsing model, which outputs a list of trees ranked by their probabilities. The scored list is treated as a random variable and the sentences whose variable has the highest entropy are selected for human annotation. Sample size was measured in TC and ranged from 100K to 700K WSJ constituents. The initial size of the unlabelled pool was 800K constituents (the 40K sentences of sections 2-21 of WSJ). A detailed comparison between the results of TE and our methods is given in Section 5.

The following works addressed the task of sample selection for statistical parsers, but in significantly different experimental setups. Becker and Osborne (2005) addressed lower performance levels of the Collins parser. Their uncertainty sampling protocol combined bagging with the TE function, achieving a 32% TC reduction for reaching a parser f-score level of 85.5%. The target sample size set contained a much smaller number of sentences ($\sim$5K) than ours. Baldridge and Osborne (2004) addressed HPSG parse selection using a feature based log-linear parser, the Redwoods corpus and committee based active learning, obtaining 80% reduction in annotation cost. Their annotation cost measure was related to the number of possible parses of the sentence. Tang et al. (2002) addressed a shallow parser trained on a semantically annotated corpus.

[1] Hwa explored several functions in the experimental setup used in the present work, and TE gave the best results.

They used an uncertainty sampling protocol, where in each iteration the sentences of the unlabelled pool are clustered using a distance measure defined on parse trees to a predefined number of clusters. The most uncertain sentences are selected from the clusters, the training taking into account the densities of the clusters. They reduced the number of training sentences required for their parser to achieve its best performance from 1300 to 400.

The importance of cognitively driven measures of sentences' syntactic complexity has been recognized by Roark et al. (2007) who demonstrated their utility for mild cognitive impairment diagnosis. Zhu et al. (2008) used a clustering algorithm for sampling the initial labeled set in an AL algorithm for word sense disambiguation and text classification. In contrast to our CBS method, they proceeded with iterative uncertainty AL selection. Melville et al. (2005) used parameter-based sample selection for a classifier in a classic active learning setting, for a task very different from ours.

Sample selection has been applied to many NLP applications. Examples include base noun phrase chunking (Ngai, 2000), named entity recognition (Tomanek et al., 2007) and multi–task annotation (Reichart et al., 2008).

3 Cognitively Driven Evaluation Measures

While the resources, capabilities and constraints of the human parser have been the subject of extensive research, different theories predict different aspects of its observed performance. We focus on structures that are widely agreed to impose a high cognitive load on the human annotator and on theories considering the cognitive resources required in parsing a complete sentence. Based on these, we derive measures for the cognitive load on the human parser when syntactically annotating a set of sentences.

Nested structures. A nested structure is a parse tree node representing a constituent created while another constituent is still being processed ('open'). The *degree* K of a nested structure is the number of such open constituents. In this paper, we enumerate the constituents in a top-down left-right order, and thus when a constituent is created, only its ancestors are processed[2]. A constituent is processed until the processing of its children is completed. For example, in Figure 1, when the constituent NP3 is created, it starts a nested structure of degree 2, since two levels of its ancestors (VP, S) are still processed. Its parent (VP) starts a nested structure of degree 1.

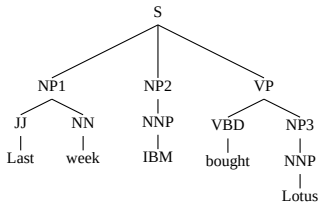

Figure 1: An example parse tree.

The difficulty of deeply nested structures for the human parser is well established in the psycholinguistics literature. We review here some of the various explanations of this phenomenon; for a comprehensive review see (Gibson, 1998).

According to the classical stack overflow theory (Chomsky and Miller, 1963) and its extension, the incomplete syntactic/thematic dependencies theory (Gibson, 1991), the human parser should track the open structures in its short term memory. When the number of these structures is too large or when the structures are nested too deeply, the short term memory fails to hold them and the sentence becomes uninterpretable.

According to the perspective shifts theory (MacWhinney, 1982), processing deeply nested structures requires multiple shifts of the annotator perspective and is thus more difficult than processing shallow structures. The difficulty of deeply nested structured has been demonstrated for many languages (Gibson, 1998).

We thus propose the total number of nested structures of degree K in a sample (TNSK) as a measure of the cognitive efforts that its annotation requires. The higher K is, the more demanding the structure.

Sentence level resources. In the psycholinguistic literature of sentence processing there are many theories describing the cognitive resources required during a complete sentence processing. These resources might be allocated during the processing of a certain word and are needed long after its constituent is closed. We briefly discuss two lines of theory, focusing on their predictions that sentences consisting of a large number of structures (e.g., con-

[2] A good review on node enumeration of the human parser in given in (Abney and Johnson, 1991).

stituents or nested structures) require more cognitive resources for longer periods.

Levelt (2001) suggested a layered model of the mental lexicon organization, arguing that when one hears or reads a sentence s/he activates word forms (lexemes) that in turn activate lemma information. The lemma information contains information about syntactic properties of the word (e.g., whether it is a noun or a verb) and about the possible sentence structures that can be generated given that word. The process of reading words and retrieving their lemma information is incremental and the lemma information for a given word is used until its syntactic structure is completed. The information about a word include all syntactic predictions, obligatory (e.g., the prediction of a noun following a determiner) and optional (e.g., optional arguments of a verb, modifier relationships). This information might be relevant long after the constituents containing the word are closed, sometimes till the end of the sentence.

Another line of research focuses on working memory, emphasizing the *activation decay* principle. It stresses that words and structures perceived during sentence processing are forgotten over time. As the distance between two related structures in a sentence grows, it is more demanding to reactivate one when seeing the other. Indeed, supported by a variety of observations, many of the theories of the human parser (see (Lewis et al., 2006) for a survey) predict that processing items towards the end of longer sentences should be harder, since they most often have to be integrated with items further back. Thus, sentences with a large number of structures impose a special cognitive load on the annotator.

We thus propose to use the number of structures (constituents or nested structures) in a sentence as a measure of its difficulty for human annotation. The measures we use for a sample (a sentence set) are the *average number of constituents* (AC) and the *average number of nested structures of degree* k (ANSK) per sentence in the set. Higher AC or ANSK values of a set imply higher annotation requirements[3].

Pschycolinguistics research makes finer observations about the human parser than those described here. A complete survey of that literature is beyond the scope of this paper. We consider the proposed measures a good approximation of some of the human parser characteristics.

[3]The correlation between the number of constituents and sentence length is very strong (e.g., correlation coefficient of 0.93 in WSJ section 0). We could use the number of words, but we prefer the number of structures since the latter better reflects the arguments made in the literature.

4 Parameter Based Sampling (PBS)

Our approach is to sample the unannotated pool with respect to the distribution of the model parameters in its sentences. In this paper, in order to compare to previous works, we apply our methods to the Collins generative parser (Collins, 1999). For any sentence s and parse tree t it assigns a probability $p(s,t)$, and finds the tree for which this probability is maximized. To do that, it writes $p(t,s)$ as a product of the probabilities of the constituents in t and decomposes the latter using the chain rule. In simplified notation, it uses:

$$p(t,s) = \prod P(S_1 \rightarrow S_2 \dots S_n) = \prod P(S_1) \cdot \dots \cdot P(S_n|\phi(S_1 \dots S_n)) \quad (1)$$

We refer to the conditional probabilities as the model *parameters*.

Cluster Based Sampling (CBS). We describe here a method for sampling subsets that leads to a parameter estimation that is similar to the parameter estimation we would get if annotating the whole unannotated set.

To do that, we randomly select M sentences from the unlabelled pool N, manually annotate them, train the parser with these sentences and parse the rest of the unlabelled pool ($G = N - M$). Using this annotation we build a syntactic vector representation for each sentence in G. We then cluster these sentences and sample the clusters with respect to their weights to preserve the distribution of the syntactic features. The selected sentences are manually annotated and combined with the group of M sentences to train the final parser. The size of this combined sample is measured when the annotation efforts are evaluated.

Denote the left hand side nonterminal of a constituent by P and the unlexicalized head of the constituent by H. The domain of P is the set of nonterminals (excluding POS tags) and the domain of H is the set of nonterminals and POS tags of WSJ. In all the parameters of the Collins parser P and H are conditioned upon. We thus use (P,H) pairs as the

features in the vector representation of each sentence in G. The i-th coordinate is given by the equation:

$$\sum_{c \in t(s)} \sum_{i} F_i(Q(c) == i) \cdot L(c) \quad (2)$$

Where c are the constituents of the sentence parse $t(s)$, Q is a function that returns the (P, H) pair of the constituent c, F_i is a predicate that returns 1 iff it is given pair number i as an argument and 0 otherwise, and L is the number of modifying non-terminals in the constituent plus 1 (for the head), counting the number of parameters that condition on (P, H). Following equation (2), the ith coordinate of the vector representation of a sentence in G contains the number of parameters that will be calculated conditioned on the ith (P, H) pair.

We use the k-means clustering algorithm, with the L_2 norm as a distance metric (MacKay, 2002), to divide vectors into clusters. Clusters created by this algorithm contain adjacent vectors in a Euclidean space. Clusters represent sentences with similar features values. To initialize k-means, we sample the initial centers values from a uniform distribution over the data points.

We do not decide on the number of clusters in advance but try to find inherent structure in the data. Several methods for estimating the 'correct' number of clusters are known (Milligan and Cooper, 1985). We used a statistical heuristic called the elbow test. We define the 'within cluster dispersion' W_k as follows. Suppose that the data is divided into k clusters $C_1 \dots C_k$ with $|C_j|$ points in the jth cluster. Let $D_t = \sum_{i,j \in C_t} d_{i,j}$ where $d_{i,j}$ is the squared Euclidean distance, then $W_k := \sum_{t=1}^{k} \frac{1}{2|C_t|} D_t$. W_k tends to decrease monotonically as k increases. In many cases, from some k this decrease flattens markedly. The heuristic is that the location of such an 'elbow' indicates the appropriate number of clusters. In our experiments, an obvious elbow occurred for 15 clusters.

k_i sentences are randomly sampled from each cluster, $k_i = D \frac{|C_i|}{\sum_j |C_j|}$, where D is the number of sentences to be sampled from G. That way we ensure that in the final sample each cluster is represented according to its size.

CMM Sampling. All of the parameters in the Collins parser are conditioned on the constituent's head word. Since word statistics are sparse, sampling from clusters created according to a lexical vector representation of the sentences does not seem promising[4].

Another way to create a sample from which the parser can extract robust head word statistics is to select a sample containing many examples of each word. More formally, we denote the words that occur in the unlabelled pool at least t times by *t-words*, where t is a parameter of the algorithm. We want to select a sample containing at least t examples of as many t-words as possible.

To select such a sample we introduce a novel optimisation problem. Our problem is a variant of the multiset multicover (MM) problem, which we call the *constrained multiset multicover* (CMM) problem. The setting of the MM problem is as follows (Hochbaum, 1997): Given a set I of m elements to be covered each b_i times, a collection of multisets $S_j \subset I$, $j \in J = \{1, \dots, n\}$ (a multiset is a set in which members' multiplicity may be greater than 1), and weights w_j, find a subcollection C of multisets that covers each $i \in I$ at least b_i times, and such that $\sum_{j \in C} w_j$ is minimized.

CMM differs from MM in that in CMM the sum of the weights (representing the desired number of sentences to annotate) is bounded, while the number of covered elements (representing the t-words) should be maximized. In our case, I is the set of words that occur at least t times in the unlabelled pool, $b_i = t, \forall i \in I$, the multisets are the sentences in that pool and $w_j = 1, \forall j \in J$.

Multiset multicover is NP-hard. However, there is a good greedy approximation algorithm for it. Define $a(s_j, i) = min(R(s_j, i), d_i)$, where d_i is the difference between b_i and the number of instances of item i that are present in our current sample, and $R(s_j, i)$ is the multiplicity of the i-th element in the multiset s_j. Define $A(s_j)$ to be the multiset containing exactly $a(s_j, i)$ copies of any element i if s_j is not already in the set cover and the empty set if it is. The greedy algorithm repeatedly adds a set minimizing $\frac{w_j}{|A(s_j)|}$. This algorithm provenly achieves an approximation ratio between $ln(m)$ and $ln(m) + 1$. In our case all weights are 1, so the algorithm would

[4] We explored CBS with several lexical features schemes and got only marginal improvement over random selection.

simply add the sentence that maximizes $A(s_j)$ to the set cover.

The problem in directly applying the algorithm to our case is that it does not take into account the desired sample size. We devised a variant of the algorithm where we use a binary tree to 'push' upwards the number of t-words in the whole batch of unannotated sentences that occurs at least t times in the selected one. Below is a detailed description. D denotes the desired number of items to sample.

The algorithm has two steps. First, we iteratively sample (without replacement) D multisets (sentences) from a uniform distribution over the multisets. In each iteration we calculate for the selected multiset its 'contribution' – the number of items that cross the threshold of t occurrences with this multiset minus the number of items that cross the t threshold without this multiset (i.e. the contribution of the first multiset is the number of t-words occurring more than t times in it). For each multiset we build a node with a key that holds its contribution, and insert these nodes in a binary tree. Insertion is done such that all downward paths are sorted in decreasing order of key values.

Second, we iteratively sample (from a uniform distribution, without replacement) the rest of the multisets pool. For each multiset we perform two steps. First, we prepare a node with a key as described above. We then randomly choose Z leaves[5] in the binary tree (if the number of leaves is smaller than Z all of the leaves are chosen). For each leaf we find the place of the new node in the path from the root to the leaf (paths are sorted in decreasing order of key values). We insert the new node to the highest such place found (if the new key is not smaller than the existing paths), add its multiset to the set of selected multisets, and remove the multiset that corresponds to the leaf of this path from the batch and the leaf itself from the binary tree. We finally choose the multisets that correspond to the highest D nodes in the tree.

An empirical demonstration of the quality of approximation that the algorithm provides is given in Figure 2. We ran our algorithm with the threshold parameter set to $t \in [2, 14]$ and counted the number of words occurring at least 5 times in the selected sample. We followed the same experimental protocol as in Section 5. The graph shows that the number of words occurring at least 5 times in a sample selected by our algorithm when $t = 5$ is significantly higher (by about a 1000) than the number of such words in a randomly selected sample and in samples selected by our algorithm with other t parameter values. We got the same pattern of results when counting words occurring at least t times for the other values of the t parameter – only the run of the algorithm with the corresponding t value created a sample with significantly higher number of words not below threshold. The other runs and random selection resulted in samples containing significantly lower number of words not below threshold.

In Section 5 we show that the parser performance when it is trained with a sample selected by CMM is significantly better than when it is trained with a randomly selected sample. Improvement is similar across the t parameter values.

[5]We tried Z values from 10 to 100 in steps of 10 and observed very similar results. We report results for $Z = 100$.

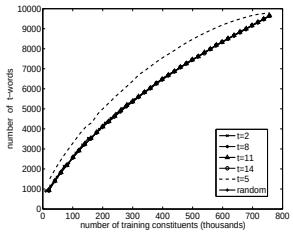

Figure 2: Number of t-words for t = 5 in samples selected by CMM runs with different values of the threshold parameter t and in a randomly selected sample. CMM with t = 5 is significantly higher. All the lines except for the line for t = 5 are unified. For clarity, we do not show all t values: their curves are also similar to the $t \neq 5$ lines.

Method	**86%**	**86.5%**	**87%**	**87.5%**	**88%**
TE	16.9% (152K)	**27.1% (183K)**	**26.9% (258K)**	14.8% (414K)	**15.8% (563 K)**
CBS	**19.6% (147K)**	16.8% (210K)	19% (286K)	**21.1% (382K)**	9% (610K)
CMM	9% (167K)	10.4% (226K)	8.9% (312K)	10.3% (433K)	14% (574K)

Table 1: Reduction in annotation cost in TC terms compared to the random baseline for tree entropy (TE), syntactic clustering (CBS) and CMM. The compared samples are the smallest samples selected by each of the methods that achieve certain f-score levels. Reduction is calculated by: $100 - 100 \times (TC_{\text{method}}/TC_{\text{random}})$.

	86%				87%				88%			
Method	TNSK (1-6)	TNSK (7-22)	ANSK (1-6)	ANSK (7-22)	TNSK (1-6)	TNSK (7-22)	ANSK (1-6)	ANSK (7-22)	TNSK (1-6)	TNSK (7-22)	ANSK (1-6)	ANSK (7-22)
TE	**34.9**%	3.6%	- 8.9%	- 61.3%	**42.2**%	14.4%	- 9.9%	- 62.7%	**25**%	8.1%	- 6.3%	- 30%
CBS	21.3%	**18.6**%	- **0.5**%	- 3.5%	19.6%	**24.2**%	- **0.3**%	- **1.8**%	8.9%	8.6 %	**0**%	- 0.3%
CMM	10.18%	8.87%	-0.82%	-**3.39%**	11%	16.22%	-0.34%	-**1.8%**	14.65%	**14.11**%	-0.02%	- **0.08**%

Table 2: Annotation cost reduction in TNSK and ANSK compared to the random baseline for tree entropy (TE), syntactic clustering (CBS) and CMM. The compared samples are the smallest selected by each of the methods that achieve certain f-score levels. Each column represents the reduction in total or average number of structures of degree 1–6 or 7–22. Reduction for each measure is calculated by: $100 - 100 \times (measure_{\text{method}}/measure_{\text{random}})$. Negative reduction is an addition. Samples with a higher reduction in a certain measure are better in terms of that measure.

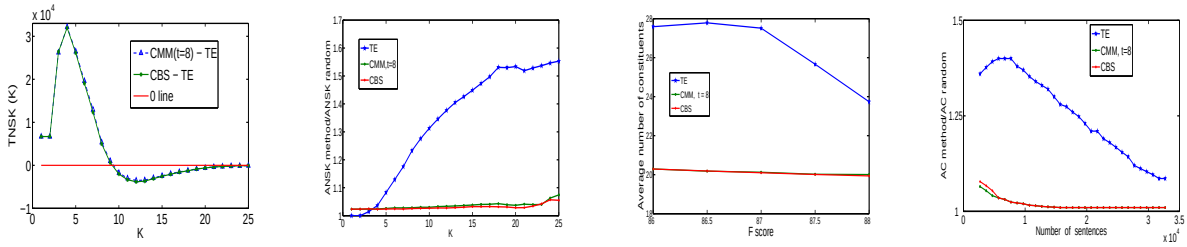

Figure 3: Left to right: First: The difference between the number of nested structures of degree K of CMM and TE and of CBS and TE. The curves are unified. The 0 curve is given for reference. Samples selected by CMM and CBS have more nested structures of degrees 1–6 and less nested structures of degrees 7–22. Results are presented for the smallest samples required for achieving f-score of 88. Similar patterns are observed for other f-score values. Second: Average number of nested structures of degree K as a function of K for the smallest sample required for achieving f-score of 88. Results for each of the methods are normalized by the average number of nested structures of degree K in the smallest randomly selected sample required for achieving f-score of 88. The sentences in CMM and CBS samples are not more complex than sentences in a randomly selected sample. In TE samples sentences are more complex. Third: Average number of constituents (AC) for the smallest sample of each of the methods that is required for achieving a given f-score. CMM and CBS samples contain sentences with a smaller number of constituents. Fourth: AC values for the samples created by the methods (normalized by AC values of a randomly selected sample). The sentences in TE samples, but not in CMM and CBS samples, are more complex than sentences in a randomly selected sample.

5 Results

Experimental setup. We used Bikel's reimplementation of Collins' parsing model 2 (Bikel, 2004). Sections 02-21 and 23 of the WSJ were stripped from their annotation. Sections 2-21 (39832 sentences, about 800K constituents) were used for training, Section 23 (2416 sentences) for testing. No development set was used. We used the gold standard POS tags in two cases: in the test section (23) in all experiments, and in Sections 02-21 in the CBS method when these sections are to be parsed in the process of vector creation. In active learning methods the unlabelled pool is parsed in each iteration and thus should be tagged with POS tags. Hwa (2004) (to whom we compare our results) used the gold standard POS tags for the same sections in her work[6]. We implemented a random baseline where sentences are uniformly selected from the unlabelled pool for annotation. For reliability we repeated each experiment with the algorithms and the random baseline 10 times, each time with different random selections (M sentences for creating syntactic tagging and k-means initialization for CBS, sentence order in CMM), and averaged the results.

Each experiment contained 38 runs. In each run a different desired sample size was selected, from 1700 onwards, in steps of 1000. Parsing performance is measured in terms of f-score

Results. We compare the performance of our CBS and CMM algorithms to the TE method (Hwa, 2004)[7], which is the only sample selection work ad-

[6]Personal communication with Hwa. Collins' parser uses an input POS tag only if it cannot tag its word using the statistics learned from the training set.

[7]Hwa has kindly sent us the samples selected by her TE. We evaluated these samples with TC and the new measures. The TC of the minimal sample she sent us needed for achieving f-score

dressing our experimental setup. Unless otherwise stated, we report the reduction in annotation cost: $100 - 100 \times (measure_{\text{method}}/measure_{\text{random}})$. CMM results are very similar for $t \in \{2, 3, \ldots, 14\}$, and presented for $t = 8$.

Table 1 presents reduction in annotation cost in TC terms. CBS achieves greater reduction for $f = 86, 87.5$, TE for $f = 86.5, 87, 88$. For $f = 88$, TE and CMM performance are almost similar. Examining the f-score vs. TC sample size over the whole constituents range (not shown due to space constraints) reveals that CBS, CMM and TE outperform random selection over the whole range. CBS and TE performance are quite similar with TE being better in the ranges of 170–300K and 520–650K constituents (42% of the 620K constituents compared) and CBS being better in the ranges of 130–170K and 300–520K constituents (44% of the range). CMM performance is worse than CBS and TE until 540K constituents. From 650K constituents on, where the parser achieves its best performance, the performance of CMM and TE methods are similar, outperforming CBS.

Table 2 shows the annotation cost reduction in ANSK and TNSK terms. TE achieves remarkable reduction in the total number of relatively shallow structures (TNSK K = 1–6). Our methods, in contrast, achieve remarkable reduction in the number of deep structures (TNSK K = 7–22)[8]. This is true for all f-score values. Moreover, the average number of nested structures per sentence, for every degree K (ANSK for every K) in TE sentences is much higher than in sentences of a randomly selected sample. In samples selected by our methods, the ANSK values are very close to the ANSK values of randomly selected samples. Thus, sentences in TE samples are much more complex than in CBS and CMM samples.

The two leftmost graphs in Figure 3 demonstrate (for the minimal samples required for f-score of 88) that these reductions hold for each K value (ANSK) and for each $K \in [7, 22]$ (TNSK) not just on the average over these K values. We observed similar results for other f-score values.

The two rightmost graphs of Figure 3 demonstrates AC results. The left of them shows that for every f-score value, the AC measure of the minimal TE sample required to achieve that f-score is higher than the AC value of PBS samples (which are very similar to the AC values of randomly selected samples). The right graph demonstrates that for every sample size, the AC value of TE samples is higher than that of PBS samples.

All AL based previous work (including TE) is iterative. In each iteration thousands of sentences are parsed, while PBS algorithms perform a single iteration. Consequently, PBS computational complexity is dramatically lower. Empirically, using a Pentium 4 2.4GHz machine, CMM requires about an hour and CBS about 16.5 hours, while the TE parsing steps alone take 662 hours (27.58 days).

6 Discussion and Future Work

We introduced novel evaluation measures: AC, TNSK and ANSK for the task of sample selection for statistical parsers. Based on the psycholinguistic literature we argue that these measures reflect aspects of the cognitive efforts of the human annotator that are not reflected by the traditional TC measure. We introduced the parameter based sample selection (PBS) approach and its CMM and CBS algorithms that do not deliberately select difficult sentences. Therefore, our intuition was that they should select a sample that leads to an accurate parameter estimation but does not contain a high number of complex structures. We demonstrated that CMM and CBS achieve results that are similar to the state of the art TE method in TC terms and outperform it when the cognitively driven measures are considered.

The measures we suggest do not provide a full and accurate description of human annotator efforts. In future work we intend to extend and refine our measures and to revise our algorithms accordingly.

We also intend to design stopping criteria for the PBS methods. These are criteria that decide when the selected sample suffices for the parser best performance and further annotation is not needed.

of 88 is different from the number reported in (Hwa, 2004). We compare our TC results with the TC result in the sample sent us by Hwa.

[8] We present results where the border between shallow and deep structures is set to be $K_{border} = 6$. For every $K_{border} \in \{7, 8, \ldots, 22\}$ TNSK reductions with CBS and CMM are much more impressive than with TE for structures whose degree is $K \in [K_{border}, 22]$.

References

Steven Abney and Mark Johnson, 1991. Memory requirements and local ambiguities of parsing strategies. *Psycholinguistic Research*, 20(3):233–250.

Daniel M. Biken, 2004. Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4):479–511.

Jason Baldridge and Miles Osborne, 2004. Active learning and the total cost of annotation. *EMNLP '04*.

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

Markus Becker, 2008. Active learning – an explicit treatment of unreliable parameters. Ph.D. thesis, The University of Edinburgh.

Noam Chomsky and George A. Miller, 1963. Finitary models of language users. In R. Duncan Luce, Robert R. Bush, and Eugene Galanter, editors, *Handbook of Mathematical Psychology*, volume II. John Wiley, New York, 419–491.

David Cohn, Les Atlas and Richard E. Ladner, 1994. Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

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

Edward Gibson, 1991. *A computational theory of human linguistic processing: memory limitations and processing breakdowns.* Ph.D. thesis, Carnegie Mellon University, Pittsburg, PA.

Edward Gibson, 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition*, 68:1–76.

Dorit Hochbaum (ed), 1997. *Approximation algorithms for NP-hard problems*. PWS Publishing, Boston.

Rebecca Hwa, 2004. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):253–276.

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

Matthew Lease and Eugene Charniak, 2005. Parsing biomedical literature. *IJCNLP '05*.

Willem J.M. Levelt, 2001. Spoken word production: A theory of lexical access. *PNAS*, 98(23):13464–13471.

Richard L. Lewis, Shravan Vasishth and Julie Van Dyke, 2006. Computational principles of working memory in sentence comprehension. *Trends in Cognitive Science*, October:447-454.

David MacKay, 2002. *Information theory, inference and learning algorithms*. Cambridge University Press.

Brian MacWhinney, 1982. The competition model. In B. MacWhinney, editor, *Mechanisms of language acquisition*. Hillsdale, NJ: Lawrence Erlbaum, 249–308.

Daniel Marcu, Wei Wang, Abdessamabad Echihabi, and Kevin Knight, 2006. SPMT: Statistical machine translation with syntactified target language phrases. *EMNLP '06.*

P. Melville, M. Saar-Tsechansky, F. Provost and R.J. Mooney, 2005. An expected utility approach to active feature-value acquisition. *5th IEEE Intl. Conf. on Data Mining '05.*

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz, 1994. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

G.W. Milligan and M.C Cooper, 1985. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 58(2):159–157.

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

Roi Reichart, Katrin Tomanek, Udo Hahn and Ari Rappoport, 2008. Multi-task active learning for linguistic annotations. *ACL '08.*

Brian Roark, Margaret Mitchell and Kristy Hollingshead, 2007. Syntactic complexity measures for detecting mild cognitive impairment. *BioNLP workshop, ACL '07*.

Min Tang, Xiaoqiang Luo, and Salim Roukos, 2002. Active learning for statistical natural language parsing. *ACL '02.*

Katrin Tomanek, Joachim Wermtre, and Udo Hahn, 2007. An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. *EMNLP '07.*

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning, 2005. Joint learning improves semantic role labeling. *ACL '05*.

Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K. Tsou, 2008. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. *COLING '08.*