Active Learning for Dependency Parsing with Partial Annotation

Zhenghua Li^{\dagger}, Min Zhang^{\dagger *}, Yue Zhang^{\dagger}, Zhanyi Liu^{\ddagger},

Wenliang Chen[†], Hua Wu[‡], Haifeng Wang[‡]

† Soochow University, Suzhou, China

{zhli13,minzhang,wlchen}@suda.edu.cn, zhangyue1107@qq.com

‡ Baidu Inc., Beijing, China

{liuzhanyi,wu_hua,wanghaifeng}@baidu.com

Abstract

Different from traditional active learning based on sentence-wise full annotation (FA). paper proposes this active learning with dependency-wise partial annotation (PA) as a finer-grained unit for dependency parsing. At each iteration, we select a few most uncertain words from an unlabeled data pool, manually annotate their syntactic heads, and add the partial trees into labeled data for parser retraining. Compared with sentence-wise FA, dependency-wise PA gives us more flexibility in task selection and avoids wasting time on annotating trivial tasks in a sentence. Our work makes the following contributions. First, we are the first to apply a probabilistic model to active learning for dependency parsing, which can 1) provide tree probabilities and dependency marginal probabilities as principled uncertainty metrics, and 2) directly learn parameters from PA based on a forest-based training objective. Second, we propose and compare several uncertainty metrics through simulation experiments on both Chinese and English. Finally, we conduct human annotation experiments to compare FA and PA on real annotation time and quality.

1 Introduction

During the past decade, supervised dependency parsing has gained extensive progress in boosting parsing performance on canonical texts, especially on texts from domains or genres similar to existing manually labeled treebanks (Koo and Collins, 2010; Zhang and Nivre, 2011). However, the

Correspondence author.





upsurge of web data (e.g., tweets, blogs, and product comments) imposes great challenges to existing parsing techniques. Meanwhile, previous research on out-of-domain dependency parsing gains little success (Dredze et al., 2007; Petrov and McDonald, 2012). A more feasible way for open-domain parsing is to manually annotate a certain amount of texts from the target domain or genre. Recently, several small-scale treebanks on web texts have been built for study and evaluation (Foster et al., 2011; Petrov and McDonald, 2012; Kong et al., 2014; Wang et al., 2014).

Meanwhile, active learning (AL) aims to reduce annotation effort by choosing and manually annotating unlabeled instances that are most valuable for training statistical models (Olsson, 2009). Traditionally, AL utilizes full annotation (FA) for parsing (Tang et al., 2002; Hwa, 2004; Lynn et al., 2012), where a whole syntactic tree is annotated for a given sentence at a time. However, as commented by Mejer and Crammer (2012), the annotation process is complex, slow, and prone to mistakes when FA is required. Particularly, annotators waste a lot of effort on labeling trivial dependencies which can be well handled by current statistical models (Flannery and Mori, 2015).

Recently, researchers report promising results with AL based on partial annotation (PA) for dependency parsing (Sassano and Kurohashi, 2010; Mirroshandel and Nasr, 2011; Majidi and Crane, 2013; Flannery and Mori, 2015). They find that smaller units rather than sentences provide more flexibility in choosing potentially informative structures to annotate.

Beyond previous work, this paper endeavors to more thoroughly study this issue, and has made substantial progress from the following perspectives.

- (1) This is the first work that applies a stateof-the-art probabilistic parsing model to AL for dependency parsing. The CRF-based dependency parser on the one hand allows us to use probabilities of trees or marginal probabilities of single dependencies for uncertainty measurement, and on the other hand can directly learn parameters from partially annotated trees. Using probabilistic models may be ubiquitous in AL for relatively simpler tasks like classification and sequence labeling, but is definitely novel for dependency parsing which is dominated by linear models with perceptron-like training.
- (2) Based on the CRF-based parser, we make systematic comparison among several uncertainty metrics for both FA and PA. Simulation experiments show that compared with using FA, AL with PA can greatly reduce annotation effort in terms of dependency number by 62.2% on Chinese and by 74.2% on English.
- (3) We build a visualized annotation platform and conduct human annotation experiments to compare FA and PA on real annotation time and quality, where we obtain several interesting observations and conclusions.

All codes, along with the data from human annotation experiments, are released at http://hlt.suda.edu.cn/~zhli for future research study.

2 Probabilistic Dependency Parsing

Given an input sentence $\mathbf{x} = w_1...w_n$, the goal of dependency parsing is to build a directed dependency tree $\mathbf{d} = \{h \curvearrowright m : 0 \le h \le n, 1 \le m \le n\}$, where $|\mathbf{d}| = n$ and $h \curvearrowright m$ represents a dependency from a *head* word h to a *modifier* word m. Figure 1 depicts a partial tree containing two dependencies.¹ In this work, we for the first time apply a probabilistic CRF-based parsing model to AL for dependency parsing. We adopt the second-order graphbased model of McDonald and Pereira (2006), which casts the problem as finding an optimal tree from a fully-connect directed graph and factors the score of a dependency tree into scores of pairs of sibling dependencies.

$$\mathbf{d}^{*} = \arg \max_{\mathbf{d} \in \mathcal{Y}(\mathbf{x})} Score(\mathbf{x}, \mathbf{d}; \mathbf{w})$$
$$Score(\mathbf{x}, \mathbf{d}; \mathbf{w}) = \sum_{\substack{(h, s, m): h \frown s \in \mathbf{d}, \\ h \frown m \in \mathbf{d}}} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, h, s, m)$$
(1)

where s and m are adjacent siblings both modifying h; f(x, h, s, m) are the corresponding feature vector; w is the feature weight vector; $\mathcal{Y}(x)$ is the set of all legal trees for x according to the dependency grammar in hand; d* is the 1-best parse tree which can be gained efficiently via a dynamic programming algorithm (Eisner, 2000). We use the state-of-the-art feature set listed in Bohnet (2010).

Under the log-linear CRF-based model, the probability of a dependency tree is:

$$p(\mathbf{d}|\mathbf{x};\mathbf{w}) = \frac{e^{Score(\mathbf{x},\mathbf{d};\mathbf{w})}}{\sum_{\mathbf{d}'\in\mathcal{Y}(\mathbf{x})} e^{Score(\mathbf{x},\mathbf{d}';\mathbf{w})}} \quad (2)$$

Ma and Zhao (2015) give a very detailed and thorough introduction to CRFs for dependency parsing.

2.1 Learning from FA

Under the supervised learning scenario, a labeled training data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{d}_i)\}_{i=1}^N$ is provided to learn w. The objective is to maximize the log likelihood of \mathcal{D} :

$$\mathcal{L}(\mathcal{D}; \mathbf{w}) = \sum_{i=1}^{N} \log p(\mathbf{d}_i | \mathbf{x}_i; \mathbf{w}) \qquad (3)$$

which can be solved by standard gradient descent algorithms. In this work, we adopt stochastic gradient descent (SGD) with L2-norm regularization for all CRF-based parsing models.²

¹In this work, we follow many previous works to focus on unlabeled dependency parsing (constructing the skeleton dependency structure). However, the proposed techniques

explored in this paper can be easily extended to the case of labeled dependency parsing.

²We borrow the implementation of SGD in CRFsuite (http://www.chokkan.org/software/crfsuite/), and use 100 sentences for a batch.

2.2 Marginal Probability of Dependencies

Marcheggiani and Artières (2014) shows that marginal probabilities of local labels can be used as an effective uncertain metric for AL for sequence labeling problems. In the case of dependency parsing, the marginal probability of a dependency is the sum of probabilities of all legal trees that contain the dependency.

$$p(h \frown m | \mathbf{x}; \mathbf{w}) = \sum_{\mathbf{d} \in \mathcal{Y}(\mathbf{x}): h \frown m \in \mathbf{d}} p(\mathbf{d} | \mathbf{x}; \mathbf{w})$$
(4)

Intuitively, marginal probability is a more principled metric for measuring reliability of a dependency since it considers all legal parses in the search space, compared to previous methods based on scores of local classifiers (Sassano and Kurohashi, 2010; Flannery and Mori, 2015) or votes of n-best parses (Mirroshandel and Nasr, 2011). Moreover, Li et al. (2014) find strong correlation between marginal probability and correctness of a dependency in cross-lingual syntax projection.

3 Active Learning for Dependency Parsing

This work adopts the standard pool-based AL framework (Lewis and Gale, 1994; McCallum and Nigam, 1998). Initially, we have a small set of labeled seed data \mathcal{L} , and a large-scale unlabeled data pool \mathcal{U} . Then the procedure works as follows.

- (1) Train a new parser on the current \mathcal{L} .
- Parse all sentences in U, and select a set of the most informative tasks U'
- (3) Manually annotate: $\mathcal{U}' \to \mathcal{L}'$
- (4) Expand labeled data: $\mathcal{L} \cup \mathcal{L}' \to \mathcal{L}$

The above steps loop for many iterations until a predefined stopping criterion is met.

The key challenge for AL is how to measure the informativeness of structures in concern. Following previous work on AL for dependency parsing, we make a simplifying assumption that if the current model is most uncertain about an output (sub)structure, the structure is most informative in terms of boosting model performance.

3.1 Sentence-wise FA

Sentence-wise FA selects K most uncertain sentences in Step (2), and annotates their whole tree structures in Step (3). In the following, we describe several uncertainty metrics and investigate

their practical effects through experiments. Given an unlabeled sentence $\mathbf{x} = w_1...w_n$, we use \mathbf{d}^* to denote the 1-best parse tree produced by the current model as in Eq. (1). For brevity, we omit the feature weight vector \mathbf{w} in the equations.

Normalized tree score. Following previous works that use scores of local classifiers for uncertainty measurement (Sassano and Kurohashi, 2010; Flannery and Mori, 2015), we use $Score(\mathbf{x}, \mathbf{d}^*)$ to measure the uncertainty of \mathbf{x} , assuming that the model is more uncertain about \mathbf{x} if \mathbf{d}^* gets a smaller score. However, we find that directly using $Score(\mathbf{x}, \mathbf{d}^*)$ always selects very short sentences due to the definition in Eq. (1). Thus we normalize the score with the sentence length n as follows.³

$$Confi(\mathbf{x}) = \frac{Score(\mathbf{x}, \mathbf{d}^*)}{n^{1.5}}$$
(5)

Normalized tree probability. The CRF-based parser allows us, for the first time in AL for dependency parsing, to directly use tree probabilities for uncertainty measurement. Unlike previous approximate methods based on k-best parses (Mirroshandel and Nasr, 2011), tree probabilities globally consider all parse trees in the search space, and thus are intuitively more consistent and proper for measuring the reliability of a tree. Our initial assumption is that the model is more uncertain about x if d* gets a smaller probability. However, we find that directly using $p(d^*|x)$ would select very long sentences because the solution space grows exponentially with sentence length. We find that the normalization strategy below works well.⁴

$$Confi(\mathbf{x}) = \sqrt[n]{p(\mathbf{d}^*|\mathbf{x})} \tag{6}$$

Averaged marginal probability. As discussed in Section 2.2, the marginal probability of a dependency directly reflects its reliability, and thus can be regarded as another global measurement besides tree probabilities. In fact, we find that the effect of sentence length is naturally handled with the following metric.⁵

$$Confi(\mathbf{x}) = \frac{\sum_{h \curvearrowright m \in \mathbf{d}^*} p(h \curvearrowright m | \mathbf{x})}{n}$$
(7)

³We have also tried replacing $n^{1.5}$ with n (still prefer short sentences) and n^2 (bias to long sentences).

⁴We have also tried $p(\mathbf{d}^*|\mathbf{x}) \times f(n)$, where $f(n) = \log n$ or $f(n) = \sqrt{n}$, but both work badly.

⁵We have also tried $\sqrt[n]{\prod_{h \frown m \in \mathbf{d}^*} p(h \frown m | \mathbf{x})}$, leading to slightly inferior results.

3.2 Single Dependency-wise PA

AL with single dependency-wise PA selects M most uncertain words from \mathcal{U} in Step (2), and annotates the heads of the selected words in Step (3). After annotation, the newly annotated sentences with partial trees \mathcal{L}' are added into \mathcal{L} . Different from the case of sentence-wise FA, \mathcal{L}' are also put back to \mathcal{U} , so that new tasks can be further chosen from them.

Marcheggiani and Artières (2014) make systematic comparison among a dozen uncertainty metrics for AL with PA for several sequence labeling tasks. We borrow three effective metrics according to their results.

Marginal probability max. Suppose $h^0 = \arg \max_h p(h \frown i | \mathbf{x})$ is the most likely head for *i*. The intuition is that the lower $p(h^0 \frown i)$ is, the more uncertain the model is on deciding the head of the token *i*.

$$Confi(\mathbf{x}, i) = p(h^0 \frown i | \mathbf{x}) \tag{8}$$

Marginal probability gap. Suppose $h^1 = \arg \max_{h \neq h^0} p(h \frown i | \mathbf{x})$ is the second most likely head for *i*. The intuition is that the smaller the probability gap is, the more uncertain the model is about *i*.

$$Confi(\mathbf{x}, i) = p(h^0 \frown i | \mathbf{x}) - p(h^1 \frown i | \mathbf{x})$$
(9)

Marginal probability entropy. This metric considers the entropy of all possible heads for i. The assumption is that the smaller the *negative* entropy is, the more uncertain the model is about i.

$$Confi(\mathbf{x}, i) = \sum_{h} p(h \frown i | \mathbf{x}) \log p(h \frown i | \mathbf{x})$$
(10)

3.3 Batch Dependency-wise PA

In the framework of single dependency-wise PA, we assume that the selection and annotation of dependencies in the same sentence are strictly independent. In other words, annotators may be asked to annotate the head of one selected word after reading and understanding a whole (sometimes partial) sentence, and may be asked to annotate another selected word in the same sentence in next AL iteration. Obviously, frequently switching sentences incurs great waste of cognitive effort,



Figure 2: An example parse forest converted from the partial tree in Figure 1.

and annotating one dependency can certainly help decide another dependency in practice.

Inspired by the work of Flannery and Mori (2015), we propose AL with batch dependencywise PA, which is a compromise between sentence-wise FA and single dependency-wise PA. In Step 2, AL with batch dependency-wise PA selects K most uncertain sentences from \mathcal{U} , and also determines r% most uncertain words from each sentence at the same time. In Step 3, annotators are asked to label the heads of the selected words in the selected sentences. We propose and experiment with the following three strategies based on experimental results of sentence-wise FA and single dependency-wise PA.

Averaged marginal probability & gap. First, select K sentences from \mathcal{U} using averaged marginal probability. Second, select r% words using marginal probability gap for each selected sentence.

Marginal probability gap. First, for each sentence in \mathcal{U} , select r% most uncertain words according to marginal probability gap. Second, select K sentences from \mathcal{U} using the averaged marginal probability gap of the selected r% words in a sentence as the uncertainty metric.

Averaged marginal probability. This strategy is the same with the above strategy, except it measures the uncertainty of a word i according to the marginal probability of the dependency pointing to i in \mathbf{d}^* , i.e., $p(j \frown i | \mathbf{x})$, where $j \frown i \in \mathbf{d}^*$.

3.4 Learning from PA

A major challenge for AL with PA is how to learn from partially labeled sentences, as depicted in Figure 1. Li et al. (2014) show that a probabilistic CRF-based parser can naturally and effectively learn from PA. The basic idea is converting a partial tree into a forest as shown in Figure 2, and using the forest as the gold-standard reference during training, also known as *ambiguous labeling* (Riezler et al., 2002; Täckström et al., 2013).

For each remaining word without head, we add all dependencies linking to it as long as the new dependency does not violate the existing dependencies. We denote the resulting forest as \mathcal{F} j, whose probability is naturally the sum of probabilities of each tree d in \mathcal{F} .

$$p(\mathcal{F}|\mathbf{x};\mathbf{w}) = \sum_{\mathbf{d}\in\mathcal{F}} p(\mathbf{d}|\mathbf{x};\mathbf{w})$$
$$= \frac{\sum_{\mathbf{d}\in\mathcal{F}} e^{Score(\mathbf{x},\mathbf{d};\mathbf{w})}}{\sum_{\mathbf{d}'\in\mathcal{Y}(\mathbf{x})} e^{Score(\mathbf{x},\mathbf{d}';\mathbf{w})}}$$
(11)

Suppose the partially labeled training data is $\mathcal{D} = \{(\mathbf{x}_i, \mathcal{F}_i)\}_{i=1}^N$. Then its log likelihood is:

$$\mathcal{L}(\mathcal{D}; \mathbf{w}) = \sum_{i=1}^{N} \log p(\mathcal{F}_i | \mathbf{x}_i; \mathbf{w})$$
(12)

Täckström et al. (2013) show that the partial derivative of the $\mathcal{L}(\mathcal{D}; \mathbf{w})$ with regard to \mathbf{w} (a.k.a the gradient) in both Equation (3) and (12) can be efficiently solved with the classic Inside-Outside algorithm.⁶

4 Simulation Experiments

We use Chinese Penn Treebank 5.1 (*CTB*) for Chinese and Penn Treebank (*PTB*) for English. For both datasets, we follow the standard data split, and convert original bracketed structures into dependency structures using Penn2Malt with its default head-finding rules. To be more realistic, we use automatic part-of-speech (POS) tags produced by a state-of-the-art CRF-based tagger (94.1% on *CTB*-test, and 97.2% on *PTB*-test, nfold jackknifing on training data), since POS tags encode much syntactic annotation. Because AL experiments need to train many parsing models, we throw out all training sentences longer than 50 to speed up our experiments. Table 1 shows the data statistics.

Following previous practice on AL with PA (Sassano and Kurohashi, 2010; Flannery and Mori, 2015), we adopt the following AL settings for both Chinese and English . The first 500 training sentences are used as the seed labeled data \mathcal{L} . In the case of FA, K = 500 new sentences

		Train	Dev	Test
Chinese	#Sentences	14,304	803	1,910
	#Tokens	318,408	20,454	50,319
English	#Sentences	39,115	1,700	2,416
	#Tokens	908,154	40,117	56,684

Table 1: Data statistics.

are selected and annotated at each iteration. In the case of single dependency-wise PA, we select and annotate M = 10,000 dependencies, which roughly correspond to 500 sentences considering that the averaged sentence length is about 22.3 in *CTB*-train and 23.2 in *PTB*-train. In the case of batch dependency-wise PA, we set K = 500, and r = 20% for Chinese and r = 10% for English, considering that the parser trained on all data achieves about 80% and 90% accuracies.

We measure parsing performance using the standard unlabeled attachment score (UAS) *including punctuation marks*. Please note that we always treat punctuation marks as ordinary words when selecting annotation tasks and calculating UAS, in order to make fair comparison between FA and PA.⁷

4.1 FA vs. Single Dependency-wise PA

First, we make comparison on the performance of AL with FA and with single dependency-wise PA.

Results on Chinese are shown in Figure 3. Following previous work, we use the number of annotated dependencies (x-axis) as the annotation cost in order to fairly compare FA and PA. We use FA with random selection as a baseline. We also draw the accuracy of the CRF-based parser trained on all training data, which can be regarded as the upper bound.

For FA, the curve of the normalized tree score intertwines with that of random selection. Meanwhile, the performance of normalized tree probability is very close to that of averaged marginal probability, and both are clearly superior to the baseline with random selection.

For PA, the difference among the three uncertainty metrics is small. The marginal probability gap clearly outperforms the other two metrics before 50,000 annotated dependencies, and remains

⁶This work focuses on projective dependency parsing. Please refer to Koo et al. (2007), McDonald and Satta (2007), and Smith and Smith (2007) for building a probabilistic nonprojective parser.

⁷Alternatively, we can exclude punctuation marks for task selection in AL with PA. Then, to be fair, we have to discard all dependencies pointing to punctuation marks in the case of FA. This makes the experiment setting more complicated.



Figure 3: FA vs. PA on CTB-dev.



Figure 4: FA vs. PA on PTB-dev.

very competitive at all other points. The marginal probability max achieves best peak UAS, and even outperforms the parser trained on all data, which can be explained by small disturbance during complex model training. The marginal probability entropy, although being the most complex metric among the three, seems inferior all the time.

It is clear that using PA can greatly reduce annotation effort compared with using FA in terms of annotated dependencies.

Results on English are shown in Figure 4. The overall findings are similar to those in Figure 3, except that the distinction among different methods is more clear. For FA, normalized tree score is consistently better than the random baseline. Normalized tree probability always outperforms normalized tree score. Averaged marginal probability performs best, except being slightly inferior to normalized tree probability in earlier stages.

For PA, it is consistent that marginal probability gap is better than marginal probability max, and marginal probability entropy is the worst.

In summary, based on the results on the de-



Figure 5: Single vs. batch dependency-wise PA on *CTB*-dev.



Figure 6: Single vs. batch dependency-wise PA on *PTB*-dev.

velopment data in Figure 3 and 4, the best AL method with PA only needs about $\frac{80,000}{318,408} = 25\%$ annotated dependencies on Chinese, and about $\frac{90,000}{908,154} = 10\%$ on English, to reach the same performance with parsers trained on all data. Moreover, the PA methods converges much faster than the FA ones, since for the same x-axis number, much more sentences (with partial trees) are used as training data for AL with PA than FA.

4.2 Single vs. Batch Dependency-wise PA

Then we make comparison on AL with single dependency-wise PA and with the more practical batch dependency-wise PA.

Results on Chinese are shown in Figure 5. We can see that the three strategies achieve very similar performance and are also very close to single dependency-wise PA. AL with batch dependency-wise PA even achieves higher accuracy before 20,000 annotated dependencies, which should be caused by the smaller active learning steps (about

2,000 dependencies at each iteration, contrasting 10,000 for single dependency-wise PA). When the training data runs out at about 7,300 dependencies, AL with batch dependency-wise PA only lags behind with single dependency-wise PA by about 0.3%, which we suppose can be reduced if larger training data is available.

Results on English are shown in Figure 6, and are very similar to those on Chinese. One tiny difference is that the marginal probability gap is slightly worse that the other two metrics. The three uncertainty metrics have very similar accuracy curves, which are also very close to the curve of single dependency-wise PA. In addition, we also try r = 20% and find that results are inferior to r = 10%, indicating that the extra 10% annotation tasks are less valuable and contributive.

4.3 Main Results on Test Data

Table 2 shows the results on test data. We compare our CRF-based parser with ZPar v 6.0^8 , a state-ofthe-art transition-based dependency parser (Zhang and Nivre, 2011). We train ZPar with default parameter settings for 50 iterations, and choose the model that performs best on dev data. We can see that when trained on all data, our CRFbased parser outperforms ZPar on both Chinese and English.

To compare FA and PA, we report the number of annotated dependencies needed under each AL strategy to achieve an accuracy lower by about 1% than the parser trained on all data.⁹

FA (best) refers to FA with averaged marginal probability, and it needs $\frac{187,123-149,051}{187,123} = 20.3\%$ less annotated dependencies than FA with random selection on Chinese, and $\frac{395,199-197,907}{395,199} = 50.0\%$ less on English.

PA (single) with marginal probability gap needs $\frac{149,051-50,958}{149,051} = 65.8\%$ less annotated dependencies than FA (best) on Chinese, and $\frac{197,907-61,448}{197,907} = 69.0\%$ less on English.

PA (batch) with marginal probability gap needs slightly more annotation than PA (single) on Chinese but slightly less annotation on English, and can reduce the amount of annotated dependencies by $\frac{149,051-56,389}{149,051} = 62.2\%$ over FA (best) on Chi-

	Chinese		English		
	#Dep labeled	UAS	#Dep labeled	UAS	
ZPar	318,408	77.97	908,154	91.45	
This parser	318,408	78.36	908,154	91.66	
FA (random)	187,123	77.43	395,199	90.67	
FA (best)	149,051	77.32	197,907	90.66	
PA (single)	50,958	77.22	61,448	90.72	
PA (batch)	56,389	77.38	51,016	90.70	

Table 2: Results on test data.

nese and by $\frac{197,907-51,016}{197,907} = 74.2\%$ on English.

5 Human Annotation Experiments

So far, we measure annotation effort in terms of the number of annotated dependencies and assume that it takes the same amount of time to annotate different words, which is obviously unrealistic. To understand whether active learning based on PA can really reduce annotation time over based on FA in practice, we build a web browser based annotation system,¹⁰ and conduct human annotation experiments on Chinese.

In this part, we use CTB 7.0 which is a newer and larger version and covers more genres, and adopt the newly proposed Stanford dependencies (de Marneffe and Manning, 2008; Chang et al., 2009) which are more understandable for annotators.¹¹ Since manual syntactic annotation is very difficult and time-consuming, we only keep sentences with length [10, 20] in order to better measure annotation time by focusing on sentences of reasonable length, which leave us 12, 912 training sentences under the official data split. Then, we use a random half of training sentences to train a CRF-based parser, and select 20% most uncertain words with marginal probability gap for each sentence of the left half.

We employ 6 postgraduate students as our annotators who are at different levels of familiarity in syntactic annotation. Before annotation, the annotators are trained for about two hours by introducing the basic concepts, guidelines, and illustrating examples. Then, they are asked to practice on the annotation system for about another two hours. Finally, all annotators are required to

⁸ http://people.sutd.edu.sg/~yue_zhang/doc/

⁹The gap 1% is chosen based on the curves on development data (Figure 3 and 4) with the following two considerations: 1) larger gap may lead to wrong impression that AL is weak; 2) smaller gap (e.g., 0.5%) cannot be reached for the worst AL method (FA: random).

¹⁰http://hlt-service.suda.edu.cn/ syn-dep-batch. Please try.

¹¹We use Stanford Parser 3.4 (2014-06-16) for constituentto-dependency structure conversion.

	Time: Sec/Dep		Annotation accuracy	
	FA	PA	FA (on 20%)	PA (diff)
Annotator #1	4.0	7.9	84.65 (73.41)	75.28 (+1.87)
Annotator #2	7.5	16.0	78.90 (72.22)	62.18 (-10.04)
Annotator #3	10.0	22.2	69.75 (59.77)	56.91 (-2.86)
Annotator #4	5.1	8.7	66.75 (49.19)	61.77 (+ 12.58)
Annotator #5	7.0	17.3	65.47 (48.50)	48.39 (-0.11)
Annotator #6	7.0	10.6	58.05 (43.28)	48.37 (+5.09)
Overall	6.7	13.6	70.36 (57.28)	59.06 (+1.78)

Table 3: Statistics of human annotation.

formally annotate the same 100 sentences. The system is programed that each sentence has 3 FA submissions and 3 PA submissions. During formal annotation, the annotators are not allowed to discuss with each other or look up any guide-line or documents, which may incur unnecessary inaccuracy in timing. Instead, the annotators can only decide the syntactic structures based on the basic knowledge of dependency grammar and one's understanding of the sentence structure. The annotation process lasts for about 5 hours. On average, each annotator completes 50 sentences with FA (763 dependencies) and 50 sentences with PA (178 dependencies).

Table 3 lists the results in descending order of an annotator's experience in syntactic annotation. The first two columns compare the time needed for annotating a dependency in seconds. On average, annotating a dependency in PA takes about twice as much time as in FA, which is reasonable considering the words to be annotated in PA may be more difficult for annotators while the annotation of some tasks in FA may be very trivial and easy. Combined with the results in Table 2, we may infer that to achieve 77.3% accuracy on CTB-test, AL with FA requires $149,051 \times 6.7 = 998,641.7$ seconds of annotation, whereas AL with batch dependency-wise PA needs $56,389 \times 13.6 =$ 766,890.4 seconds. Thus, we may roughly say that AL with PA can reduce annotation time over FA by $\frac{998,641.7-766,890.4}{998,641.7} = 23.2\%.$

We also report annotation accuracy according to the gold-standard Stanford dependencies converted from bracketed structures.¹² Overall, the accuracy of FA is 70.36 - 59.06 = 11.30% higher

than that of PA, which should be due to the trivial tasks in FA. To be more fair, we compare the accuracies of FA and PA on the same 20% selected difficult words, and find that annotators exhibit different responses to the switch. Annotator #4 achieve 12.58% higher accuracy when under PA than under FA. The reason may be that under PA, annotators can be more focused and therefore perform better on the few selected tasks. In contrast, some annotators may perform better under FA. For example, annotation accuracy of annotator #2 increases by 10.04% when switching from PA to FA, which may be due to that FA allows annotators to spend more time on the same sentence and gain help from annotating easier tasks. Overall, we find that the accuracy of PA is 59.06 - 57.28 = 1.78%higher than that of FA, indicating that PA actually can improve annotation quality.

6 Related Work

Recently, AL with PA attracts much attention in sentence-wise natural language processing such as sequence labeling and parsing. For sequence labeling, Marcheggiani and Artières (2014) systematically compare a dozen uncertainty metrics in token-wise AL with PA (without comparison with FA), whereas Settles and Craven (2008) investigate different uncertainty metrics in AL with FA. Li et al. (2012) propose to only annotate the most uncertain word boundaries in a sentence for Chinese word segmentation and show promising results on both simulation and human annotation experiments. All above works are based on CRFs and make extensive use of sequence probabilities and token marginal probability.

In parsing community, Sassano and Kurohashi (2010) select bunsetsu (similar to phrases) pairs with smallest scores from a local classifier, and let annotators decide whether the pair composes a dependency. They convert partially annotated instances into local dependency/non-dependency classification instances to help a simple shiftreduce parser. Mirroshandel and Nasr (2011) select most uncertain words based on votes of nbest parsers, and convert partial trees into full trees by letting a baseline parser perform constrained decoding in order to preserve partial annotation. Under a different query-by-committee AL framework, Majidi and Crane (2013) select most uncertain words using a committee of diverse parsers, and convert partial trees into full trees by letting

¹²An anonymous reviewer commented that the direct comparison between an annotator's performance on PA and FA based on accuracy may be misleading since the FA and PA sentences for one annotator are mutually exclusive.

the parsers of committee to decide the heads of remaining tokens. Based on a first-order (pointwise) Japanese parser, Flannery and Mori (2015) use scores of a local classifier for task selection, and treat PA as dependency/non-dependency instances (Flannery et al., 2011). Different from above works, this work adopts a state-of-the-art probabilistic dependency parser, uses more principled tree probabilities and dependency marginal probabilities for uncertainty measurement, and learns from PA based on a forest-based training objective which is more theoretically sound.

Most previous works on AL with PA only conduct simulation experiments. Flannery and Mori (2015) perform human annotation to measure true annotation time. A single annotator is employed to annotate for two hours alternating FA and PA (33% batch) every fifteen minutes. Beyond their initial expectation, they find that the annotation time per dependency is nearly the same for FA and PA (different from our findings) and gives a few interesting explanations.

Under a non-AL framework, Mejer and Crammer (2012) propose an interesting light feedback scheme for dependency parsing by letting annotators decide the better one from top-2 parse trees produced by the current parsing model.

Hwa (1999) pioneers the idea of using PA to reduce manual labeling effort for constituent grammar induction. She uses a variant Inside-Outside re-estimation algorithm (Pereira and Schabes, 1992) to induce a grammar from PA. Clark and Curran (2006) propose to train a Combinatorial Categorial Grammar parser using partially labeled data only containing predicate-argument dependencies. Tsuboi et al. (2008) extend CRFbased sequence labeling models to learn from incomplete annotations, which is the same with Marcheggiani and Artières (2014). Li et al. (2014) propose a CRF-based dependency parser that can learn from partial tree projected from sourcelanguage structures in the cross-lingual parsing scenario. Mielens et al. (2015) propose to impute missing dependencies based on Gibbs sampling in order to enable traditional parsers to learn from partial trees.

7 Conclusions

This paper for the first time applies a state-ofthe-art probabilistic model to AL with PA for dependency parsing. It is shown that the CRF- based parser can on the one hand provide tree probabilities and dependency marginal probabilities as principled uncertainty metrics and on the other hand elegantly learn from partially annotated data. We have proposed and compared several uncertainty metrics through simulation experiments, and show that AL with PA can greatly reduce the amount of annotated dependencies by 62.2%on Chinese 74.2% on English. Finally, we conduct human annotation experiments on Chinese to compare PA and FA on real annotation time and quality. We find that annotating a dependency in PA takes about 2 times long as in FA. This suggests that AL with PA can reduce annotation time by 23.2% over with FA on Chinese. Moreover, the results also indicate that annotators tend to perform better under PA than FA.

For future work, we would like to advance this study in the following directions. The first idea is to combine uncertainty and representativeness for measuring informativeness of annotation targets in concern. Intuitively, it would be more profitable to annotate instances that are both difficult for the current model and representative in capturing common language phenomena. Second, we so far assume that the selected tasks are equally difficult and take the same amount of effort for human annotators. However, it is more reasonable that human are good at resolving some ambiguities but bad at others. Our plan is to study which syntactic structures are more suitable for human annotation, and balance informativeness of a candidate task and its suitability for human annotation. Finally, one anonymous reviewer comments that we may use automatically projected trees (Rasooli and Collins, 2015; Guo et al., 2015; Ma and Xia, 2014) as the initial seed labeled data, which is cheap and interesting.

Acknowledgments

The authors would like to thank the anonymous reviewers for the helpful comments. We also thank Junhui Li and Chunyu Kit for reading our paper and giving many good suggestions. Particularly, Zhenghua is very grateful to many of his students: Fangli Lu, Qiuyi Yan, and Yue Zhang build the annotation system; Jiayuan Chao, Wei Chen, Ziwei Fan, Die Hu, Qingrong Xia, and Yue Zhang participate in data annotation. This work was supported by National Natural Science Foundation of China (Grant No. 61502325, 61525205, 61572338).

References

- Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING*, pages 89–97.
- Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009. Discriminative reordering with Chinese grammatical relations features. In *Proceedings of the Third Workshop* on Syntax and Structure in Statistical Translation (SSST-3) at NAACL HLT 2009, pages 51–59.
- Stephen Clark and James Curran. 2006. Partial training for a lexicalized-grammar parser. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 144–151.
- Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.
- Mark Dredze, John Blitzer, Partha Pratim Talukdar, Kuzman Ganchev, João Graca, and Fernando Pereira. 2007. Frustratingly hard domain adaptation for dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL* 2007.
- Jason Eisner. 2000. Bilexical grammars and their cubic-time parsing algorithms. In *Advances in Probabilistic and Other Parsing Technologies*, pages 29–62.
- Daniel Flannery and Shinsuke Mori. 2015. Combining active learning and partial annotation for domain adaptation of a japanese dependency parser. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 11–19.
- Daniel Flannery, Yusuke Miayo, Graham Neubig, and Shinsuke Mori. 2011. Training dependency parsers from partially annotated corpora. In *Proceedings of IJCNLP*, pages 776–784.
- Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Joseph Le Roux, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. From news to comment: Resources and benchmarks for parsing the language of web 2.0. In *Proceedings of IJCNLP*, pages 893– 901.
- Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of ACL*, pages 1234–1244.
- Rebecca Hwa. 1999. Supervised grammar induction using training data with limited constituent information. In *Proceedings of ACL*, pages 73–79.
- Rebecca Hwa. 2004. Sample selection for statistical parsing. *Computional Linguistics*, 30(3):253–276.

- Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. 2014. A dependency parser for tweets. In *Proceedings of EMNLP*, pages 1001–1012.
- Terry Koo and Michael Collins. 2010. Efficient thirdorder dependency parsers. In ACL, pages 1–11.
- Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. Structured prediction models via the matrix-tree theorem. In *Proceedings* of *EMNLP-CoNLL*, pages 141–150.
- David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12.
- Shoushan Li, Guodong Zhou, and Chu-Ren Huang. 2012. Active learning for Chinese word segmentation. In *Proceedings of COLING 2012: Posters*, pages 683–692.
- Zhenghua Li, Min Zhang, and Wenliang Chen. 2014. Soft cross-lingual syntax projection for dependency parsing. In *COLING*, pages 783–793.
- Teresa Lynn, Jennifer Foster, Mark Dras, and Elaine U1 Dhonnchadha. 2012. Active learning and the irish treebank. In *Proceedings of ALTA*.
- Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of ACL*, pages 1337–1348.
- Xuezhe Ma and Hai Zhao. 2015. Probabilistic models for high-order projective dependency parsing. *Arxiv*, abs/1502.04174.
- Saeed Majidi and Gregory Crane. 2013. Active learning for dependency parsing by a committee of parsers. In *Proceedings of IWPT*, pages 98–105.
- Diego Marcheggiani and Thierry Artières. 2014. An experimental comparison of active learning strategies for partially labeled sequences. In *Proceedings of EMNLP*, pages 898–906.
- Andrew McCallum and Kamal Nigam. 1998. Employing EM and pool-based active learning for text classification. In *Proceedings of ICML*, pages 350–358.
- Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL*, pages 81–88.
- Ryan McDonald and Giorgio Satta. 2007. On the complexity of non-projective data-driven dependency parsing. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 121– 132.

- Avihai Mejer and Koby Crammer. 2012. Training dependency parser using light feedback. In *Proceedings of NAACL*.
- Jason Mielens, Liang Sun, and Jason Baldridge. 2015. Parse imputation for dependency annotations. In *Proceedings of ACL-IJCNLP*, pages 1385–1394.
- Seyed Abolghasem Mirroshandel and Alexis Nasr. 2011. Active learning for dependency parsing using partially annotated sentences. In *Proceedings* of the 12th International Conference on Parsing Technologies, pages 140–149.
- Fredrik Olsson. 2009. A literature survey of active machine learning in the context of natural language processing. Technical report, Swedish Institute of Computer Science.
- Fernando Pereira and Yves Schabes. 1992. Insideoutside reestimation from partially bracketed corpora. In *Proceedings of the Workshop on Speech and Natural Language (HLT)*, pages 122–127.
- Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *Notes* of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL).
- Mohammad Sadegh Rasooli and Michael Collins. 2015. Density-driven cross-lingual transfer of dependency parsers. In *Proceedings of EMNLP*, pages 328–338.
- Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. III Maxwell, and Mark Johnson. 2002. Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 271–278.
- Manabu Sassano and Sadao Kurohashi. 2010. Using smaller constituents rather than sentences in active learning for japanese dependency parsing. In *Proceedings of ACL*, pages 356–365.
- Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of EMNLP*, pages 1070–1079.
- David A. Smith and Noah A. Smith. 2007. Probabilistic models of nonprojective dependency trees. In *Proceedings of EMNLP-CoNLL*, pages 132–140.
- Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of NAACL*, pages 1061–1071.
- Min Tang, Xiaoqiang Luo, and Salim Roukos. 2002. Active learning for statistical natural language parsing. In *Proceedings of ACL*, pages 120–127.

- Yuta Tsuboi, Hisashi Kashima, Hiroki Oda, Shinsuke Mori, and Yuji Matsumoto. 2008. Training conditional random fields using incomplete annotations. In *Proceedings of COLING*, pages 897–904.
- William Yang Wang, Lingpeng Kong, Kathryn Mazaitis, and William W Cohen. 2014. Dependency parsing for weibo: An efficient probabilistic logic programming approach. In *Proceedings of EMNLP*, pages 1152–1158.
- Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of ACL*, pages 188–193.