Enhancing Active Learning for Semantic Role Labeling via Compressed Dependency Trees

Chenhua Chen, Alexis Palmer, and Caroline Sporleder
Computational Linguistics and Phonetics
Saarland University, Saarbrücken, Germany
ch.chenua@googlemail.com, {apalmer, csporled}@coli.uni-sb.de

Abstract

This paper explores new approaches to active learning (AL) for semantic role labeling (SRL), focusing in particular on combining typical informativity-based sampling strategies with a novel measure of representativeness based on compressed dependency trees (CDTs). In essence, the compressed representation encodes the target predicate and the key dependents of the verb complex in the sentence. We first present our method for producing CDTs from the output of an existing dependency parser. The compressed trees are used as features for training a supervised SRL system. Second, we present a study of AL for SRL. We investigate a number of different sample selection strategies, and the best results are achieved by incorporating CDTs for example selection based on both informativity and representativeness. We show that our approach can reduce by up to 50% the amount of training data needed to attain a given level of performance.

1 Introduction

The focus of this paper is active learning for semantic role labeling, a little-studied intersection of two rather substantial bodies of work.

One aim of active learning (AL) is to reduce the number of labeled training instances required to reach a given performance level using supervised machine learning techniques. This is accomplished by allowing the learner to guide the selection of examples to be annotated and added to the training set; at each iteration the learner queries for the example (or set of examples) that will be most informative to its present state. AL is an attractive idea for natural language processing (NLP) because of its potential to dramatically reduce the need for expensive expert annotation, and it has been successfully applied in various areas of natural language processing (Tang et al., 2002; Settles and Craven, 2008), including named entity recognition (Shen et al., 2004),text classification (Yang et al., 2009), image retrieval (Zhou, 2006), part-of-speech tagging (Ringger et al., 2007), morpheme glossing (Baldridge and Palmer, 2009), and syntactic parsing (Hwa, 2004; Osborne and Baldridge, 2004).

The problems of scarce annotated data and the expense of annotating new data are at least as relevant for semantic role labeling (SRL) as for the above-mentioned areas of NLP. Existing work on automatic SRL usually explores supervised machine learning approaches to mark the semantic roles of predicates automatically by training classifiers using large annotated corpora.[1] Although such approaches can achieve reasonably good performance, annotating a large corpus is still expensive and time consuming. Moreover, the performance of trained classifiers may degrade remarkably when they are applied to out-of-domain data (Johansson and Nugues, 2008a). There is very little work on AL for SRL (e.g. Roth and Small (2006)), although much interesting work has been done with semi-supervised and unsupervised approaches to the problem (Grenager and Manning, 2006; Fürstenau and Lapata, 2009; Lang and Lapata, 2010; Titov and Klementiev, 2011, among others).

In this paper we explore the use of compressed dependency trees (CDTs) as features for supervised semantic role labeling and, most importantly, as a way of measuring how representative an individual instance is of the input data. We then incorporate representativeness as part of the metric used for sample selection in active learning. The

[1] For recent work on SRL, see, among others: (Das et al., 2010; Hajič et al., 2009; Surdeanu et al., 2008; Carreras and Màrquez, 2005; Baker et al., 2007).

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 183–191,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

compressed dependency trees encode the target predicate and the key dependents of the verb complex in a sentence. As illustrated in Section 3, the structural relationships defined by the compressed dependency trees well encapsulate key features used in automatic SRL.

For a more complete picture of the potential for AL with respect to SRL, we investigate a set of strategies designed to select the most **informative** training examples. We further develop a more effective approach to select training examples concerning both their informativity and **representativeness**. We use the compressed dependency trees to measure the similarity of two sentences, and select the training examples with a higher priority which are more informative and representative among the unlabeled sentences in the pool. The experimental results show that our approaches can reduce up to 50% of training examples compared to traditional supervised learning solutions.

We begin with a brief description of the semantic role labeling task and our supervised learning model. Section 3 presents our method for compressing dependency tree representations, followed by the active learning model, including definitions of all sampling strategies investigated in this work (Section 4). Experiments and results are presented and discussed in Section 5 and Section 6. We end with related work (Section 7) and brief conclusions.

2 Semantic Role Labeling

Parsing the semantic argument structure of a sentence involves identification and disambiguation of target predicates as well as identification and labeling of their arguments. Because our focus is on the active learning more so than on the semantic role labeling itself, we address only the argument labeling stage of the process, assuming that predicates and argument spans alike have already been identified and correctly labeled.

Broadly speaking, there are two different styles of semantic parsing and semantic role labeling (SRL): those based on FrameNet-style analysis (Ruppenhofer et al., 2006) and those using PropBank-style analysis (Palmer et al., 2005). This work takes the PropBank approach, which considers only verbal predicates and is strongly tied to syntactic structure. In (1), for example, the two arguments of the predicate *idolize* are labeled as *Arg0* and *Arg1*.

(1) [John]$_{Arg0}$ idolizes [his sister]$_{Arg1}$.

In this text, we refer to each argument to be labeled, together with its target predicate, as an **instance**; the sentence in (1) contains two instances.

2.1 Supervised Learning Model

The aim of the current work is not to surpass state-of-the-art performance on semantic role labeling. Therefore, although state-of-the-art semantic role labelers are freely available, we chose to implement our own labeler in order to have more control over the underlying machinery. This allows straightforward access to the predicted probability of outputs, which is crucial for the informativity-based selection strategies in Section 4. In addition, compressed dependency trees (Section 3) serve as features for our labeler as well as guiding sample selection in the active learning experiments.

In our study, we applied an L1-regularized[2] logistic regression model (Lee et al., 2006) for labeling instances, using the liblinear package (Lin et al., 2007) to build one classifier per label. There are 6 core and 13 non-core argument labels in PropBank annotations. Thus our SRL system is a suite of binary classifiers, and we then use the one-versus-all method (Duda et al., 2001) to assign labels to each instance.

2.2 Data and Features

We used the version of PropBank provided for the CoNLL-2008 SRL shared task (Surdeanu et al., 2008). A test set of 500 randomly selected sentences was constructed at the outset of the project; this was used only for evaluation of both supervised and active learning models. In all AL experiments, we simulate the oracle by hiding and then uncovering gold-standard labels.

The CoNLL-2008 data set includes both gold-standard dependency parses and automatic dependency parses from the Malt parser (Nivre and Hall, 2005). We use a combination of features taken directly from the gold-standard parses,[3] features derived from the Malt parses, and features from the output of the Stanford dependency parser (de

[2] Note that logistic regression is used together with a regularized term to avoid the overfitting problem by penalizing the complexity of the trained model. Generally, the regularized term is defined as a function of the learned parameters over the weights. The L1 regularization, also called lasso penalty, is used to penalize both large and small weights.

[3] In ongoing work, we replace gold-standard parses with more realistic automatic parses.

Table 1: Three feature groups: CoNLL basic, CoNLL derived, and from additional parser

FEATURE TYPE	EXPLANATION/EXAMPLE
Part of Speech	JJR, JJS, LS, CD, etc.
Head word	Head words of predicate and argument
isNEG	Instance includes NOT or NEVER
Argument position	Before or after predicate
Argument chunk position	Beginning or end of corresponding chunk
Lemma of argument	Lemma of argument whose dependency role is PRD or DIR
Lemma context	Two words before and after argument
Cue words	DIR ('up', 'toward', 'forward', 'along') REC ('self' as suffix) PRD ('as', 'as if') CAU ('because', 'why', 'as a result of')
Voice of predicate	Active or passive
Dependency relation of predicate and argument	LOC, TMP, etc. 1) Sbj*, obj* are defined as: Sbj* ← Obj_Passive Sbj* ← LGS_passive Sbj* ← Active_vt_sbj Obj* ← Sbj_Passive Obj* ← Sbj_VI (intransitive verb) Obj* ← Obj_Active

Predicate Properties	VT = 1; transitive
	VI = 2; intransitive
	TO_IM=3; begins with 'to'
	V_ Adj = 4; verb followed by adjective words (e.g. 'sounds good', 'looks pretty')
	PV = 5; phrasal verb (e.g. 'pick up')

Verb Complex	e.g. "has not been set" in figure 1
Acomp	adjectival complement
Advmod	adverbial modifier
Infmod	infinitival modifier
Rcmod	relative clause modifier
Rel	relative (word introducing an rcmod)
Xsbj	controlling subject
Iobj	indirect object
Advcl	adverbial clause modifier
Prep_to,Prep_in, Prep_for, Prep_with	Prepositional phrases with 'to', 'in', 'for', 'with'

Marneffe et al., 2006). To apply the logistic regression model, the features are represented in a binary fashion. The features are described in Table 1, in three groups separated by double lines. The derived features, including a heuristically-identified verb complex and altered dependency labels, are described in more detail in Section 3.

We use cross-validation on the training data to select for each individual classifier the subset of features most relevant for that label. In feature selection, features are ranked based on their Fisher score calculated using the training data set (as in Duda et al. (2001)).

3 Dependency Tree Compression

Given a sentence, the task of dependency parsing is to identify the head word and its corresponding dependents and to classify their functional relationships according to a set of dependency relations (e.g., subject, modifier). Thus, a dependency tree of a sentence encodes the dependency relation between the head words and their dependents. It has been reported that SRL can benefit from phrase-structure and dependency-based syntactic parsing (Hacioglu, 2004; Johansson and Nugues, 2008b; Pradhan et al., 2005). At the same time, much of the structural and relational information represented in a dependency tree is not relevant for the SRL task.

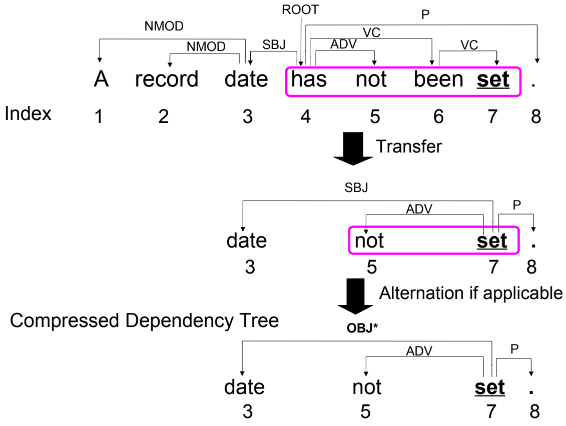


Figure 1: Producing compressed dependency tree

We use a **compressed dependency tree** (CDT) to encode just the relationships between a target predicate and the key dependents of the verb complex. The new tree is always rooted in the target predicate, which often means resetting the root from an auxiliary or other finite main verb. We generate the CDT from the output of an existing dependency parser through the process described in a simplified form below, using the example sentence in Fig. 1.

1. Fix target predicate (e.g. *set*) as root of CDT.
2. Identify the verb chain to which the target predicate belongs; this group of tokens will now be treated as the **verb complex**. The verb chain is produced by collecting elements connected by relevant dependency relations (VC, IM, CONJ), stopping when a ROOT node, a subordinate clause (SUB), or a verbal OBJ node is encountered.
3. Collect direct dependents of each word in the new verb complex; set these as dependents of the target predicate in the CDT, transferring the dependency relation to the target predicate. (e.g. *date* is a dependent of *have*).
4. Negation, modal verbs, and other main verbs in the verb complex also become dependents of the root predicate in the CDT. In some cases of 'new' dependency relations introduced by the tree compression process, we use output from the Stanford parser to complement the dependency relations found in the gold-standard data.

5. Heuristically determine voice of clause and alter some CDT dependency labels(e.g. SBJ_PASSIVE becomes OBJ*); these are the asterisk-marked relations in Table. 1.

For example, in (2):

(2) At the same time, the government did not want to appear to favor GM by allowing a minority stake that might preclude a full bid by Ford.

the verb complex is {*did, n't, want, appear, favor*}. The subject phrase *the government*, originally a dependent of *did*, becomes a dependent of the new three-verb predicate {*want, appear, favor*}; the negation word *n't* is a dependent of the target predicate *want*.

4 Active Learning

This section provides some background on the active learning process, as well as detailing the various sampling strategies we investigate.

4.1 The basic model

In this study we apply a standard active learning model (Settles, 2010; Lewis and Gale, 1994) to the task of semantic role labeling. Algorithm 1 illustrates this model as we use it.[4]

Algorithm 1 Active learning for SRL.

```
1: Randomly select initial seed of labeled instances;
2: Add initial seed to the training data;
3: Apply logistic regression model to train system of classifiers, one for each label;
4: while number of instances in training data is less than $X$ do
5:     Randomly select pool of $Y$ unlabeled sentences;
6:     Select a sentence or sentences from the unlabeled pool according to a given selection strategy;
7:     Ask oracle to label the selected unlabeled sentence;
8:     Add instances from selected sentence to training data;
9:     Re-train system using the updated training data;
10:    Use system to label test data, record accuracy;
11: end while
```

Much recent work in AL has to do with Step 6 of Algorithm 1, designing and refining selection strategies. The main selection criterion used to date has been **informativity**, measuring how much a training example can help to reduce the uncertainty of a statistical model. A less-frequently considered criterion, especially in AL for NLP, is **representativeness**, or how well a training example represents the overall input patterns of the unlabeled data.

While some results from AL are robust across different datasets and even different tasks, it is clear that there is no single approach to AL that is suitable for all situations (Tomanek and Olsson, 2009). Because there is very little previous work on AL for the task of semantic role labeling, we do not assume previous solutions but rather investigate a number of different strategies.

4.2 Informativity

Informativity is exploited in our approaches in terms of uncertainty, which is measured based on how confidently the system labels instances and, by extension, sentences. The lower the confidence on labeling a particular sentence, the more uncertainty is assigned to the sentence. At each iteration, then, we select from the unlabeled pool the single sentence with the greatest uncertainty. We compare 4 different scoring functions for measuring the system's certainty (CER) regarding an unlabeled sentence. These are presented below as INF1-INF4.

Let s represent an unlabeled sentence with instances $i = 1$ to n. Given a set of binary classifiers, one each for labels $y = 1$ to m, let $p_{i,y}$ be the probability of i being labeled as y. Finally, P is a pool of unlabeled sentences. At each iteration, we select the single $s \in P$ with the lowest value for CER.

RAND: Random selection. Random selection (randomly select an unlabeled sentence $s \in P$) serves as a strong baseline in active learning.

INF1: Average uncertainty. After labeling each instance in a sentence with the most-likely predicted label, we calculate uncertainty for the sentence as the average of the classifiers' confidence in assigning the predicted labels. Let $Top(i) = p_{i,y_k}$, where $\forall h \neq k, p_{i,y_k} > p_{i,y_h}$; $CER(s) = (\sum_{j=1}^{n} Top(i_j))/n$.

INF2: Average uncertainty variance. Our second informativity-based strategy evaluates the uncertainty of the labeling for an instance using the variance of the confidence for each instance. A smaller variance implies that it is more difficult for the system to differentiate between possible label assignments for the instance. We then calculate sentence uncertainty as the average variance for

[4] Recall that each sentence contains one or more instances.

all instances. Let $AVG(i) = (\sum_{k=1}^{m} p_{i,y_k})/m$, $VAR(i) = \sum_{k=1}^{m}(p_{i,y_k} - AVG(i))^2/(m-1)$; $CER(s) = \sum_{j=1}^{n} VAR(i_j)/n$.

INF3: Average top-2 Margin. The intuition behind this approach is that the top 2 most confident labels are likely to be more informative than other labels. Therefore, we only select the two most likely labels to calculate uncertainty.[5] Let $Margin(i) = p_{i,y_{k_1}} - p_{i,y_{k_2}}$, where $p_{i,y_{k_1}} > p_{i,y_{k_2}} \wedge \forall h \neq k_1, k_2, p_{i,y_{k_2}} > p_{i,k_h}$; $CER(s) = (\sum_{j=1}^{n} Margin(i_j))/n$.

INF4: Most top-2 Margin Instances. Finally, we further extend the approach of INF3 by selecting the sentence which has the greatest number of instances with a small margin between the top 2 labels (which means that the sentence is more uncertain than other sentences). Let Q be a set of instances with the top-2 margin less than a small threshold (i.e., $Margin(i) \leq 0.1$). $CER(s)$ is defined as the inverse of the number of instances of s that are in Q (i.e. $1/\#$ *qualifying instances*). Ties are resolved by random selection.

4.3 Representativeness

A disadvantage of selecting examples based only on informativity is the tendency of the learner to query outliers (Settles, 2010). It has therefore been proposed (Dredze and Crammer, 2008; Settles and Craven, 2008) to temper such selection strategies with a notion of relevance or representativeness. Ours is the first work to use such a combined strategy for SRL. We measure the representativeness of unlabeled sentences based on sentence similarity, taking two different approaches: cosine similarity, and a measure based on CDTs.

COS: Cosine Similarity. Given two sentences s and s', let $i_1, i_2, \ldots, i_m$, and $i'_1, i'_2, \ldots, i'_n$ be their instances, respectively. The similarity of the two sentences, denoted as $similarity(s, s')$, is defined as $\sum_{j=1}^{m}\sum_{k=1}^{n} sim(i_j, i'_k)$, where $sim(i_j, i'_k)$ is the similarity between the instances i_j and i'_k, defined as the cosine of the two feature vectors.[6] For purposes of comparison, we use the same formulation of COS as Settles and Craven (2008).

Given a pool P of unlabeled sentences, for every unlabeled sentence $s \in P$, the representativeness of the sentence, denoted as $rep(s)$, is measured as the sum of the similarity between the sentence and all the other sentences in the pool, that is, $rep(s) = \sum sim(s, s')$, where $s' \in P \wedge s' \neq s$.

COS evaluates the similarity of two sentences based on the cosine of their instances. This may not be accurate enough because the instances include more information than the relationships between the target predicate and the key dependents of the verb complex in the sentence. Therefore, we exploit the compressed dependency trees as a metric to evaluate the similarity between two sentences, as illustrated below:

CDT: Compressed Dependency Trees. For target predicate p, let (p, r_i, a_i) be the edges of the CDT rooted in p, where a_i is an argument and r_i is the dependency relationship between p and a_i. We call two edges similar if **all** of p, r, and a meet their respective similarity criteria. Two predicates are considered to be similar if they have the same value for the PREDICATE PROPERTIES feature as defined in Table 1 (e.g. both are transitive verbs). Two relations are considered to be similar if they have the same dependency relation label (e.g. SBJ, TMP, MOD, etc.). Finally, two arguments are considered to be similar if they share the same coarse-grained part-of-speech tag.

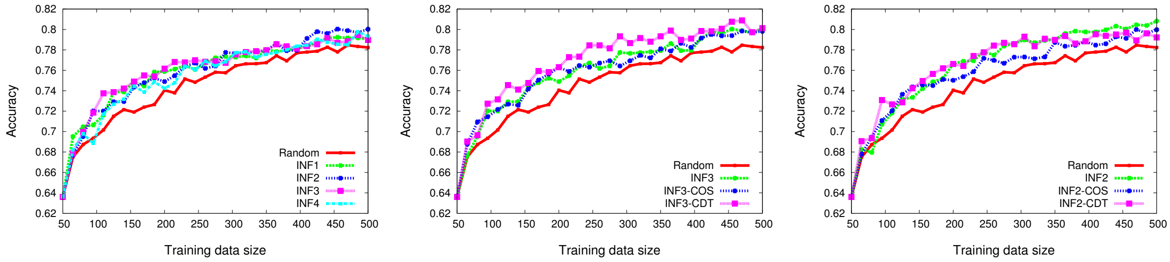
Given a pool P of unlabeled sentences, for every unlabeled sentence $s \in P$, the representativeness of the sentence, denoted as $rep(s)$, is defined as $n_{similar}$, representing the number of edges in the pool that are similar to the edges of the CDT for s. Intuitively, the larger the number of similar CDT edges in the unlabeled pool, the more representative the sentence is overall of the input data.

4.4 Combining Informativity and Representativeness

The final step in our model is to define a selection strategy that incorporates *both* selection criteria. We define the priority of selecting a sentence as $priority(s) = \alpha \times rep(s) - (1-\alpha) \times CER(s)$. Given a pool P, we select the single $s \in P$ with the highest value for $priority(s)$. This approach is very similar to the information density (ID) approach of Settles and Craven (2008); the key difference is in the balance between the two criteria. Ours is a linear combination; ID instead multiplies informativity by a weighted measure of rep-

[5] Note that in the binary classification case, INF3 is equivalent to INF1.

[6] Features are extracted from CDTs rather than full sentences, reducing to some extent the appearance of noisy information (e.g. stop words). Whether this can be further reduced by a modified implementation of COS is a question for future work.



(a) Informativity-only strategies

(b) INF3 with COS & CDT

(c) INF2 with COS &CDT

Figure 2: Combining informativity and representativeness.

resentativeness.

5 Experimental Setup

To evaluate our approach to AL for SRL, we investigate three different questions. First, which informativity strategy is most appropriate for the task? Second, which representativeness measure works best? And third, how shall we weight the trade-off between the two selection criteria?

All of our active learning experiments share some characteristics. First, we randomly select a seed of 50 instances from the labeled training data. The seed set, as well as the test data, are kept consistent across all experimental conditions. In each iteration of the training-selection cycle (see Algorithm 1), a new unlabeled pool (n=500) is selected, and from that pool a single example is labeled by the oracle and added to the training set. We stop once 500 examples have been labeled.

To evaluate the effectiveness of each strategy, we tested the classifier in each interaction, and measured the accuracy of the predicted labels. The accuracy measure is defined as the number of correct labelings divided by the total number of labelings in the test data. Results are presented as the average over 20 runs.

To investigate the influence of representativeness, we run the same experiment with all cross-combinations of {INF1,INF2,INF3,INF4} and {COS,CDT}. For weighting the two criteria, we use both information density (ID) as defined in Settles and Craven (2008) and our *priority* metric (Section 4.4) with α set at 0.3, 0.5, and 0.7.

6 Results and Discussion

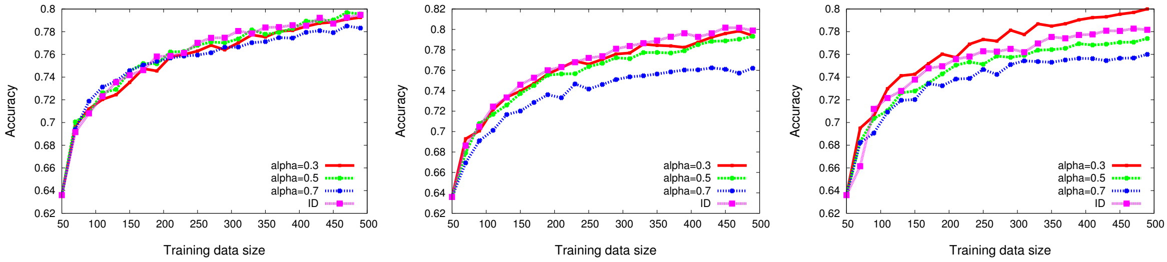
In this section, we analyze and discuss the experimental results. The gains achieved by AL can be measured in a number of different ways; first, we plot number of labeled training examples against system accuracy (Figure 2 and Figure 3). The figures presented here stop at 500 training examples, with averaged accuracies in the range of 80%. For comparison, the fully-supervised system when trained on *20000* instances performed at 89.71%. Second, we calculate the percent reduction in error of each strategy compared to the random selection baseline (Table 2), following Melville and Mooney (2004). Because most gains from AL happen early in the learning curve, we consider performance at two different points.

6.1 Informativity-based Strategies

Fig. 2a shows the expected result that the four informativity-based strategies outperform the random selection baseline. INF3 performs best early in the learning curve, but is overtaken by INF2 at the end of our curve. To reach the accuracy achieved by the four informativity strategies at the halfway point (250 training instances), RAND needs 100-150 additional instances.

6.2 Informativity plus Representativeness

Fig. 2b shows the result of combining the informativity (INF3) and representativeness (both COS and CDT). As illustrated in Section 6.1, INF3 outperforms the other informativity-based strategies. However, we see that Fig. 2b shows combining CDT with INF3 achieves a better performance than using INF3 only ($\alpha = 0.3$); representativeness improves performance, outperforming RAND by approximately 250 training instances. For INF3, COS is a less effective measure of representativeness. This may be because the feature vectors for the training instances share too much information, including stop words and a large number of 0-valued features, to make them easily differentiated. As a result, the most representative sentence selected using COS may not reflect the real simi-



(a) INF3 with COS

(b) INF3 with CDT

(c) INF2 with CDT

Figure 3: Trade-off between informativity and representativeness.

larity of the sentences. In CDT, we choose only the structural relation between the predicate and its arguments to measure the similarity between sentences. As a result, the sentences selected using CDT are more representative than that of using COS, as confirmed by the result in Fig. 2b.

We also applied the solution of combining informativity and representativeness (4.3) to other informativity-based strategies. However, the advantage the combined solution for other strategies is less obvious than for INF3. For example, Fig. 2c shows the result of combining INF2 ($\alpha = 0.3$) with both COS and CDT. The result shows that the combined solution with CDT performs slightly better than using INF2 only when the number of training instances is less than 200. However, when the number of instances is larger than 350, the solution of using INF2 only achieves a higher accuracy than the combined solution. This may be due to a conflict between the two selection criteria. In any event, there is clearly a trade-off between informativity and representativeness, and results are influenced by the details of the manner of combining the two.

The results of other INF/REP combinations are presented in Table 2, in terms of their reduction in error compared to random selection.

6.3 Weighting the two criteria

Finally, we set α with different values (i.e., 0.3, 0.5 and 0.7) to investigate how the trade-off between informativity and representativeness may affect the SRL performance. We also compare our solution to the information density solution proposed by *et al.* (Settles and Craven, 2008) (denoted as ID) multiplies the informativity and representativeness instead of summing them. Here we display only the results of INF2 and INF4 combining with CDT in Fig. 3. Other combinations share a similar pattern with these results and their error reduction percentage can be found in Table. 2. Fig. 3a and Fig. 3b compare the two representativity measures for INF3, as the best overall result was achieved by INF3 in combination with CDT. We see that parameter tuning seems to be more influential for the CDT measure than for the COS measure.

Fig. 3c shows how parameter tuning affects INF2; $\alpha = 0.3$ has a higher accuracy than that of 0.5 and 0.7. We can observe that when $\alpha = 0.3$, our solution (INF2) has a better performance than that of ID. However, regarding the combination of INF4 and CDT, ID performs better (no graph; see 2. Note that the INF4 selects the sentences which has greatest number of instances with a small margin. Then representativeness of the sentences within the margin was calculated. In other word, the combination was done step by step not in parallel as the other combination. Therefore, the combination of INF4 and CDT accounts for informativity prior to representativeness; this may be why ID is more successful.

In general, the balance and trade-offs between the two criteria deserve further investigation.

7 Related Work

Much research efforts have been devoted to statistical machine learning methodologies for SRL (Bjkelund et al., 2009; Gildea and Jurafsky, 2002; Shi et al., 2009; Johansson and Nugues, 2008a; Lang and Lapata, 2010; Pradhan et al., 2008; Fürstenau and Lapata, 2009; Titov and Klementiev, 2011, among others). For example, Johansson *et al.* (Johansson and Nugues, 2008a) applied logistic regression with L2 norm to dependency-based SRL. Similarly, we also use logistic regression to train the classifier with a probabilistic explanation. However, we use L1 normed

Table 2: Percentage error reduction over RAND(200 / 500 examples)

	NOREP	COS	CDT	COS-ID	CDT-ID
INF1	6.56 / 5.18	3.68 / -0.86	2.60 / -0.74	7.43 / 6.45	6.44 / 6.60
INF2	5.12 / 8.31	5.51 / 5.37	7.74 / 8.19	7.21 / 5.67	3.49 / 2.24
INF3	5.07 / 5.54	6.13 / 5.52	8.15 / 9.54	5.94 / 5.72	5.65 / 7.18
INF4	7.37 / 5.79	1.41 / 2.01	-0.01 / -5.08	2.29 / 2.85	3.31 / 3.29

logistic regression due to its desirable property that can result in few nonzero feature weights. This allows us to select the most important features from an otherwise very large feature set.

Roth *et al.* (Roth and Small, 2006) proposed a margin based active learning framework for structured output and experiment on SRL task. They defined structured output by constraining the relations among class labels, e.g., one predicate only has one of the labels. The classification problem is defined via constraints among output labels. The most uncertain instances are selected to satisfy predefined constraints. Rather than a structured relation between output labels, our work exploits the structure of the sentences themselves via compressed dependency trees.

In the area of sentence similarity measurement, most current work focuses on semantic similarity (Haghighi et al., 2005; Tang et al., 2002; Shen and Lapata, 2007). We define similarity between sentences in terms of the nodes and edges in the dependency tree instead of semantic/lexical similarity of the sentences. We are interested in the structure of a sentence and how it is constructed due to the need of SRL tasks. Wang and Neumann (2007) use a similar sort of compressed dependency tree comprised of keywords and collapsed dependency relations to calculate the semantic similarity of sentences for the textual entailment task. Under their approach, dependency relations themselves are collapsed; we keep the specific dependency relations and collapse the trees, aiming for structural rather than semantic similarity.

In addition, Filippova *et al.* (Filippova and Strube, 2008) proposed to compress a sentence using dependency trees and take the importance of words as weight. They found compressed dependency tree can better ensure the grammaticality of the sentences to preserve the same lexical meaning as much as possible. In our work, we are more interested in the explicit dependency relation of predicate-argument pairs. Our goal is to apply compressed dependency tree to extract explicit relation between predicate and argument as precise as possible for SRL purpose. Therefore, we construct the compressed tree by identifying predicate-argument units and then re-linking them if there exist dependency relation among them. Consequently, most of the nodes in our compressed tree are predicates and arguments.

8 Conclusions

This paper investigates the use of active learning for semantic role labeling. To improve the learning accuracy and reduce the size of training set, compressed dependency trees are exploited as features. Strategies to select informative unlabeled sentences are proposed. Moreover, the compressed dependency trees are also utilized as a criterion to measure the representativeness of unlabeled sentences. A solution to select unlabeled sentences combining both informativeness and representativeness is developed. The experimental results show that our solution can save up to 50% on a small training data set compared to the supervised learning solution.

Possibilities for future work include exploring the use of constraints on label outputs, implementation of entropy-based informativity metrics, and perhaps combining COS andCDT for measuring representativeness. Another potentially promising direction is to employ multi-kernel based methods as a structure-oriented similarity measurement.

Acknowledgments

Our thanks to the anonymous reviewers for their valuable commentary and suggestions, and to Ivan Titov for invaluable, insightful discussions and feedback. This research has been funded by the German Research Foundation (DFG) under the MMCI Cluster of Excellence.

References

C. Baker, M. Ellsworth, K. Erk. 2007. SemEval-2007 task 19: Frame semantic structure extraction. In *Proc. of SemEval-2007*.

J. Baldridge, A. Palmer. 2009. How well does active learning actually work? time-based evaluation of cost-reduction strategies for language documentation. In *Proc. of EMNLP 2009*.

A. Bjkelund, L. Hafdell, P. Nugues, 2009. *Multilingual Semantic Role Labeling*, 43–48. 2009.

X. Carreras, L. Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proc. of CoNLL-2005*.

D. Das, N. Schneider, D. Chen, N. A. Smith. 2010. Probabilistic frame-semantic parsing. In *Proc. of NAACL-HLT 2010*.

M.-C. de Marneffe, B. MacCartney, C. D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. of LREC 2006*.

M. Dredze, K. Crammer. 2008. Active learning with confidence. In *Proc. of ACL 2008*.

R. Duda, P. Hart, D. Stork. 2001. *Pattern classification*, volume 2. Wiley.

K. Filippova, M. Strube. 2008. Dependency tree based sentence compression. In *Proc. of INLG 2008*.

H. Fürstenau, M. Lapata. 2009. Semi-supervised semantic role labeling. In *Proc. of EACL 2009*.

D. Gildea, D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28.

T. Grenager, C. Manning. 2006. Unsupervised discovery of a statistical verb lexicon. In *Proc. of EMNLP 2006*.

K. Hacioglu. 2004. Semantic role labeling using dependency trees. In *Proc. of COLING 2004*.

A. D. Haghighi, A. Y. Ng, C. D. Manning. 2005. Robust textual inference via graph matching. In *Proc. of HLT 2005*.

J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, P. Straňák, M. Surdeanu, N. Xue, Y. Zhang. 2009. The CoNLL 2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL 2009*.

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

R. Johansson, P. Nugues. 2008a. Dependency-based semantic role labeling of propbank. In *Proc. of EMNLP 2008*.

R. Johansson, P. Nugues. 2008b. The effect of syntactic representation on semantic role labeling. In *Proc. of COLING 2008*.

J. Lang, M. Lapata. 2010. Unsupervised induction of semantic roles. In *Proc. of HLT 2010*.

S. Lee, H. Lee, P. Abbeel, A. Ng. 2006. Efficient L1 regularized logistic regression. In *Proc. of the Ntl. Conf. on AI*, volume 21.

D. D. Lewis, W. A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proc. of SIGIR 1994*.

C.-J. Lin, R. C. Weng, S. S. Keerthi. 2007. Trust region newton methods for large-scale logistic regression. In *Proc. of ICML 2007*.

P. Melville, R. J. Mooney. 2004. Diverse ensembles for active learning. In *Proc. of ICML 2004*.

J. Nivre, J. Hall. 2005. Maltparser: A language-independent system for data-driven dependency parsing. In *Proc. of the TLT 2005*.

M. Osborne, J. Baldridge. 2004. Ensemble-based active learning for parse selection. In *Proc. of HLT-NAACL 2004*.

M. Palmer, D. Gildea, P. Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–105.

S. Pradhan, W. Ward, K. Hacioglu, J. H. Martin, D. Jurafsky. 2005. Semantic role labeling using different syntactic views. In *Proc. of ACL 2005*.

S. S. Pradhan, W. Ward, J. H. Martin. 2008. Towards robust semantic role labeling. *Computational Linguistics*, 34:289–310.

E. Ringger, P. McClanahan, R. Haertel, G. Busby, M. Carmen, J. Carroll, D. Lonsdale. 2007. Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Proc. of the Linguistic Annotation Workshop*.

D. Roth, K. Small. 2006. Active learning with perceptron for structured output. In *ICML Workshop on Learning in Structured Output Spaces*.

J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, C. R. Johnson, J. Scheffczyk. 2006. FrameNet II: Extended Theory and Practice.

B. Settles, M. Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proc. of EMNLP 2008*.

B. Settles. 2010. Active learning literature survey. Technical Report Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2010.

D. Shen, M. Lapata. 2007. Using semantic roles to improve question answering. In *Proc. of EMNLP-2007*.

D. Shen, J. Zhang, J. Su, G. Zhou, C.-L. Tan. 2004. Multi-criteria-based active learning for named entity recognition. In *Proc. of ACL 2004*.

H. Shi, G. Zhou, P. Qian, X. Li. 2009. Semantic role labeling based on dependency tree with multi-features. In *Proc. of IJCBS 2009*.

M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, J. Nivre. 2008. The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proc. of CoNLL 2008*.

M. Tang, X. Luo, S. Roukos. 2002. Active learning for statistical natural language parsing. In *Proc. of ACL 2002*.

I. Titov, A. Klementiev. 2011. A bayesian model for unsupervised semantic parsing. In *Proc. of ACL 2011*.

K. Tomanek, F. Olsson. 2009. A Web Survey on the Use of Active learning to support annotation of text data. In *Proc. of AL-NLP workshop, NAACL HLT 2009*.

R. Wang, G. Neumann. 2007. Recognizing textual entailment using sentence similarity based on dependency tree skeletons. In *Proc. of RTE 2007*.

B. Yang, J.-T. Sun, T. Wang, Z. Chen. 2009. Effective multi-label active learning for text classification. In *Proc. of KDD 2009*.

Z.-H. Zhou. 2006. Learning with unlabeled data and its application to image retrieval. In *Proc. of PRICAI 2006*.