Fully Unsupervised Core-Adjunct Argument Classification

Omri Abend*
Institute of Computer Science
The Hebrew University
omria01@cs.huji.ac.il

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

Abstract

The core-adjunct argument distinction is a basic one in the theory of argument structure. The task of distinguishing between the two has strong relations to various basic NLP tasks such as syntactic parsing, semantic role labeling and subcategorization acquisition. This paper presents a novel unsupervised algorithm for the task that uses no supervised models, utilizing instead state-of-the-art syntactic induction algorithms. This is the first work to tackle this task in a fully unsupervised scenario.

1 Introduction

The distinction between core arguments (henceforth, cores) and adjuncts is included in most theories on argument structure (Dowty, 2000). The distinction can be viewed syntactically, as one between obligatory and optional arguments, or semantically, as one between arguments whose meanings are predicate dependent and independent. The latter (cores) are those whose function in the described event is to a large extent determined by the predicate, and are obligatory. Adjuncts are optional arguments which, like adverbs, modify the meaning of the described event in a predictable or predicate-independent manner.

Consider the following examples:

1. The surgeon operated [on his colleague].
2. Ron will drop by [after lunch].
3. Yuri played football [in the park].

The marked argument is a core in 1 and an adjunct in 2 and 3. Adjuncts form an independent semantic unit and their semantic role can often be inferred independently of the predicate (e.g., [after lunch] is usually a temporal modifier). Core roles are more predicate-specific, e.g., [on his colleague] has a different meaning with the verbs 'operate' and 'count'.

Sometimes the same argument plays a different role in different sentences. In (3), [in the park] places a well-defined situation (Yuri playing football) in a certain location. However, in "The troops are based [in the park]", the same argument is obligatory, since being based requires a place to be based in.

Distinguishing between the two argument types has been discussed extensively in various formulations in the NLP literature, notably in PP attachment, semantic role labeling (SRL) and subcategorization acquisition. However, no work has tackled it yet in a fully unsupervised scenario. Unsupervised models reduce reliance on the costly and error prone manual multi-layer annotation (POS tagging, parsing, core-adjunct tagging) commonly used for this task. They also allow to examine the nature of the distinction and to what extent it is accounted for in real data in a theory-independent manner.

In this paper we present a fully unsupervised algorithm for core-adjunct classification. We utilize leading fully unsupervised grammar induction and POS induction algorithms. We focus on prepositional arguments, since non-prepositional ones are generally cores. The algorithm uses three measures based on different characterizations of the core-adjunct distinction, and combines them using an ensemble method followed by self-training. The measures used are based on selectional preference, predicate-slot collocation and argument-slot collocation.

We evaluate against PropBank (Palmer et al., 2005), obtaining roughly 70% accuracy when evaluated on the prepositional arguments and more than 80% for the entire argument set. These results are substantially better than those obtained by a non-trivial baseline.

* Omri Abend is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

Section 2 discusses the core-adjunct distinction. Section 3 describes the algorithm. Sections 4 and 5 present our experimental setup and results.

2 Core-Adjunct in Previous Work

PropBank. PropBank (PB) (Palmer et al., 2005) is a widely used corpus, providing SRL annotation for the entire WSJ Penn Treebank. Its core labels are predicate specific, while adjunct (or modifiers under their terminology) labels are shared across predicates. The adjuncts are subcategorized into several classes, the most frequent of which are locative, temporal and manner[1].

The organization of PropBank is based on the notion of diathesis alternations, which are (roughly) defined to be alternations between two subcategorization frames that preserve meaning or change it systematically. The frames in which each verb appears were collected and sets of alternating frames were defined. Each such set was assumed to have a unique set of roles, named 'role-set'. These roles include all roles appearing in any of the frames, except of those defined as adjuncts.

Adjuncts are defined to be optional arguments appearing with a wide variety of verbs and frames. They can be viewed as fixed points with respect to alternations, i.e., as arguments that do not change their place or slot when the frame undergoes an alternation. This follows the notions of optionality and compositionality that define adjuncts.

Detecting diathesis alternations automatically is difficult (McCarthy, 2001), requiring an initial acquisition of a subcategorization lexicon. This alone is a challenging task tackled in the past using supervised parsers (see below).

FrameNet. FrameNet (FN) (Baker et al., 1998) is a large-scale lexicon based on frame semantics. It takes a different approach from PB to semantic roles. Like PB, it distinguishes between core and non-core arguments, but it does so for each and every frame separately. It does not commit that a semantic role is consistently tagged as a core or a non-core across frames. For example, the semantic role 'path' is considered core in the 'Self Motion' frame, but as non-core in the 'Placing' frame. Another difference is that FN does not allow any type of non-core argument to attach to a given frame. For instance, while the 'Getting' frame allows a 'Duration' non-core argument, the 'Active Perception' frame does not.

PB and FN tend to agree in clear (prototypical) cases, but to differ in others. For instance, both schemes would tag "Yuri played football [in the park]" as an adjunct and "The commander placed a guard [in the park]" as a core. However, in "He walked [into his office]", the marked argument is tagged as a directional adjunct in PB but as a 'Direction' core in FN.

Under both schemes, non-cores are usually confined to a few specific semantic domains, notably time, place and manner, in contrast to cores that are not restricted in their scope of applicability. This approach is quite common, e.g., the COBUILD English grammar (Willis, 2004) categorizes adjuncts to be of manner, aspect, opinion, place, time, frequency, duration, degree, extent, emphasis, focus and probability.

Semantic Role Labeling. Work in SRL does not tackle the core-adjunct task separately but as part of general argument classification. Supervised approaches obtain an almost perfect score in distinguishing between the two in an in-domain scenario. For instance, the confusion matrix in (Toutanova et al., 2008) indicates that their model scores 99.5% accuracy on this task. However, adaptation results are lower, with the best two models in the CoNLL 2005 shared task (Carreras and Màrquez, 2005) achieving 95.3% (Pradhan et al., 2008) and 95.6% (Punyakanok et al., 2008) accuracy in an adaptation between the relatively similar corpora WSJ and Brown.

Despite the high performance in supervised scenarios, tackling the task in an unsupervised manner is not easy. The success of supervised methods stems from the fact that the predicate-slot combination (slot is represented in this paper by its preposition) strongly determines whether a given argument is an adjunct or a core (see Section 3.4). Supervised models are provided with an annotated corpus from which they can easily learn the mapping between predicate-slot pairs and their core/adjunct label. However, induction of the mapping in an unsupervised manner must be based on inherent core-adjunct properties. In addition, supervised models utilize supervised parsers and POS taggers, while the current state-of-the-art in unsupervised parsing and POS tagging is considerably worse than their supervised counterparts.

This challenge has some resemblance to un-

[1] PropBank annotates modals and negation words as modifiers. Since these are not arguments in the common usage of the term, we exclude them from the discussion in this paper.

supervised detection of multiword expressions (MWEs). An important MWE sub-class is that of phrasal verbs, which are also characterized by verb-preposition pairs (Li et al., 2003; Sporleder and Li, 2009) (see also (Boukobza and Rappoport, 2009)). Both tasks aim to determine semantic compositionality, which is a highly challenging task.

Few works addressed unsupervised SRL-related tasks. The setup of (Grenager and Manning, 2006), who presented a Bayesian Network model for argument classification, is perhaps closest to ours. Their work relied on a supervised parser and a rule-based argument identification (both during training and testing). Swier and Stevenson (2004, 2005), while addressing an unsupervised SRL task, greatly differ from us as their algorithm uses the VerbNet (Kipper et al., 2000) verb lexicon, in addition to supervised parses. Finally, Abend et al. (2009) tackled the argument identification task alone and did not perform argument classification of any sort.

PP attachment. PP attachment is the task of determining whether a prepositional phrase which immediately follows a noun phrase attaches to the latter or to the preceding verb. This task's relation to the core-adjunct distinction was addressed in several works. For instance, the results of (Hindle and Rooth, 1993) indicate that their PP attachment system works better for cores than for adjuncts.

Merlo and Esteve Ferrer (2006) suggest a system that jointly tackles the PP attachment and the core-adjunct distinction tasks. Unlike in this work, their classifier requires extensive supervision including WordNet, language-specific features and a supervised parser. Their features are generally motivated by common linguistic considerations. Features found adaptable to a completely unsupervised scenario are used in this work as well.

Syntactic Parsing. The core-adjunct distinction is included in many syntactic annotation schemes. Although the Penn Treebank does not explicitly annotate adjuncts and cores, a few works suggested mapping its annotation (including function tags) to core-adjunct labels. Such a mapping was presented in (Collins, 1999). In his Model 2, Collins modifies his parser to provide a core-adjunct prediction, thereby improving its performance.

The Combinatory Categorial Grammar (CCG) formulation models the core-adjunct distinction explicitly. Therefore, any CCG parser can be used as a core-adjunct classifier (Hockenmaier, 2003).

Subcategorization Acquisition. This task specifies for each predicate the number, type and order of obligatory arguments. Determining the allowable subcategorization frames for a given predicate necessarily involves separating its cores from its allowable adjuncts (which are not framed). Notable works in the field include (Briscoe and Carroll, 1997; Sarkar and Zeman, 2000; Korhonen, 2002). All these works used a parsed corpus in order to collect, for each predicate, a set of hypothesized subcategorization frames, to be filtered by hypothesis testing methods.

This line of work differs from ours in a few aspects. First, all works use manual or supervised syntactic annotations, usually including a POS tagger. Second, the common approach to the task focuses on syntax and tries to identify the entire frame, rather than to tag each argument separately. Finally, most works address the task at the verb type level, trying to detect the allowable frames for each type. Consequently, the common evaluation focuses on the quality of the allowable frames acquired for each verb type, and not on the classification of specific arguments in a given corpus. Such a token level evaluation was conducted in a few works (Briscoe and Carroll, 1997; Sarkar and Zeman, 2000), but often with a small number of verbs or a small number of frames. A discussion of the differences between type and token level evaluation can be found in (Reichart et al., 2010).

The core-adjunct distinction task was tackled in the context of child language acquisition. Villavicencio (2002) developed a classifier based on preposition selection and frequency information for modeling the distinction for locative prepositional phrases. Her approach is not entirely corpus based, as it assumes the input sentences are given in a basic logical form.

The study of prepositions is a vibrant research area in NLP. A special issue of *Computational Linguistics*, which includes an extensive survey of related work, was recently devoted to the field (Baldwin et al., 2009).

3 Algorithm

We are given a (predicate, argument) pair in a test sentence, and we need to determine whether the argument is a core or an adjunct. Test arguments are assumed to be correctly bracketed. We are allowed to utilize a training corpus of raw text.

3.1 Overview

Our algorithm utilizes statistics based on the (predicate, slot, argument head) (PSH) joint distribution (a slot is represented by its preposition). To estimate this joint distribution, PSH samples are extracted from the training corpus using unsupervised POS taggers (Clark, 2003; Abend et al., 2010) and an unsupervised parser (Seginer, 2007). As current performance of unsupervised parsers for long sentences is low, we use only short sentences (up to 10 words, excluding punctuation). The length of test sentences is not bounded. Our results will show that the training data accounts well for the argument realization phenomena in the test set, despite the length bound on its sentences. The sample extraction process is detailed in Section 3.2.

Our approach makes use of both aspects of the distinction – obligatoriness and compositionality. We define three measures, one quantifying the obligatoriness of the slot, another quantifying the selectional preference of the verb to the argument and a third that quantifies the association between the head word and the slot irrespective of the predicate (Section 3.3).

The measures' predictions are expected to coincide in clear cases, but may be less successful in others. Therefore, an ensemble-based method is used to combine the three measures into a single classifier. This results in a high accuracy classifier with relatively low coverage. A self-training step is now performed to increase coverage with only a minor deterioration in accuracy (Section 3.4).

We focus on prepositional arguments. Non-prepositional arguments in English tend to be cores (e.g., in more than 85% of the cases in PB sections 2–21), while prepositional arguments tend to be equally divided between cores and adjuncts. The difficulty of the task thus lies in the classification of prepositional arguments.

3.2 Data Collection

The statistical measures used by our classifier are based on the (predicate, slot, argument head) (PSH) joint distribution. This section details the process of extracting samples from this joint distribution given a raw text corpus.

We start by parsing the corpus using the Seginer parser (Seginer, 2007). This parser is unique in its ability to induce a bracketing (unlabeled parsing) from raw text (without even using POS tags) with strong results. Its high speed (thousands of words per second) allows us to use millions of sentences, a prohibitive number for other parsers.

We continue by tagging the corpus using Clark's unsupervised POS tagger (Clark, 2003) and the unsupervised Prototype Tagger (Abend et al., 2010)[2]. The classes corresponding to prepositions and to verbs are manually selected from the induced clusters[3]. A preposition is defined to be any word which is the first word of an argument and belongs to a prepositions cluster. A verb is any word belonging to a verb cluster. This manual selection requires only a minute, since the number of classes is very small (34 in our experiments). In addition, knowing what is considered a preposition is part of the task definition itself.

Argument identification is hard even for supervised models and is considerably more so for unsupervised ones (Abend et al., 2009). We therefore confine ourselves to sentences of length not greater than 10 (excluding punctuation) which contain a single verb. A sequence of words will be marked as an argument of the verb if it is a constituent that does not contain the verb (according to the unsupervised parse tree), whose parent is an ancestor of the verb. This follows the pruning heuristic of (Xue and Palmer, 2004) often used by SRL algorithms.

The corpus is now tagged using an unsupervised POS tagger. Since the sentences in question are short, we consider every word which does not belong to a closed class cluster as a head word (an argument can have several head words). A closed class is a class of function words with relatively few word types, each of which is very frequent. Typical examples include determiners, prepositions and conjunctions. A class which is not closed is open. In this paper, we define closed classes to be clusters in which the ratio between the number of word tokens and the number of word types ex-

[2] Clark's tagger was replaced by the Prototype Tagger where the latter gave a significant improvement. See Section 4.

[3] We also explore a scenario in which they are identified by a supervised tagger. See Section 4.

ceeds a threshold T[4].

Using these annotation layers, we traverse the corpus and extract every (predicate, slot, argument head) triplet. In case an argument has several head words, each of them is considered as an independent sample. We denote the number of times that a triplet occurred in the training corpus by $N(p, s, h)$.

3.3 Collocation Measures

In this section we present the three types of measures used by the algorithm and the rationale behind each of them. These measures are all based on the PSH joint distribution.

Given a (predicate, prepositional argument) pair from the test set, we first tag and parse the argument using the unsupervised tools above[5]. Each word in the argument is now represented by its word form (without lemmatization), its unsupervised POS tag and its depth in the parse tree of the argument. The last two will be used to determine which are the head words of the argument (see below). The head words themselves, once chosen, are represented by the lemma. We now compute the following measures.

Selectional Preference (SP). Since the semantics of cores is more predicate dependent than the semantics of adjuncts, we expect arguments for which the predicate has a strong preference (in a specific slot) to be cores.

Selectional preference induction is a well-established task in NLP. It aims to quantify the likelihood that a certain argument appears in a certain slot of a predicate. Several methods have been suggested (Resnik, 1996; Li and Abe, 1998; Schulte im Walde et al., 2008).

We use the paradigm of (Erk, 2007). For a given predicate slot pair (p, s), we define its preference to the argument head h to be:

$$SP(p, s, h) = \sum_{h' \in Heads} Pr(h'|p, s) \cdot sim(h, h')$$

$$Pr(h|p, s) = \frac{N(p, s, h)}{\Sigma_{h'} N(p, s, h')}$$

$sim(h, h')$ is a similarity measure between argument heads. $Heads$ is the set of all head words. This is a natural extension of the naive (and sparse) maximum likelihood estimator $Pr(h|p, s)$, which is obtained by taking $sim(h, h')$ to be 1 if $h = h'$ and 0 otherwise.

The similarity measure we use is based on the slot distributions of the arguments. That is, two arguments are considered similar if they tend to appear in the same slots. Each head word h is assigned a vector where each coordinate corresponds to a slot s. The value of the coordinate is the number of times h appeared in s, i.e. $\Sigma_{p'} N(p', s, h)$ (p' is summed over all predicates). The similarity measure between two head words is then defined as the cosine measure of their vectors.

Since arguments in the test set can be quite long, not every open class word in the argument is taken to be a head word. Instead, only those appearing in the top level (depth = 1) of the argument under its unsupervised parse tree are taken. In case there are no such open class words, we take those appearing in depth 2. The selectional preference of the whole argument is then defined to be the arithmetic mean of this measure over all of its head words. If the argument has no head words under this definition or if none of the head words appeared in the training corpus, the selectional preference is undefined.

Predicate-Slot Collocation. Since cores are obligatory, when a predicate persistently appears with an argument in a certain slot, the arguments in this slot tends to be cores. This notion can be captured by the $(predicate, slot)$ joint distribution. We use the Pointwise Mutual Information measure (PMI) to capture the slot and the predicate's collocation tendency. Let p be a predicate and s a slot, then:

$$PS(p, s) = PMI(p, s) = log \frac{Pr(p, s)}{Pr(s) \cdot Pr(p)} =$$

$$= log \frac{N(p, s) \Sigma_{p', s'} N(p', s')}{\Sigma_{s'} N(p, s') \Sigma_{p'} N(p', s)}$$

Since there is only a meager number of possible slots (that is, of prepositions), estimating the $(predicate, slot)$ distribution can be made by the maximum likelihood estimator with manageable sparsity.

In order not to bias the counts towards predicates which tend to take more arguments, we define here $N(p, s)$ to be the number of times the (p, s) pair occurred in the training corpus, irrespective of the number of head words the argument had (and not e.g., $\Sigma_h N(p, s, h)$). Argu-

[4] We use sections 2–21 of the PTB WSJ for these counts, containing 0.95M words. Our T was set to 50.

[5] Note that while current unsupervised parsers have low performance on long sentences, arguments, even in long sentences, are usually still short enough for them to operate well. Their average length in the test set is 5.1 words.

ments with no prepositions are included in these counts as well (with $s = NULL$), so not to bias against predicates which tend to have less non-prepositional arguments.

Argument-Slot Collocation. Adjuncts tend to belong to one of a few specific semantic domains (see Section 2). Therefore, if an argument tends to appear in a certain slot in many of its instances, it is an indication that this argument tends to have a consistent semantic flavor in most of its instances. In this case, the argument and the preposition can be viewed as forming a unit on their own, independent of the predicate with which they appear. We therefore expect such arguments to be adjuncts.

We formalize this notion using the following measure. Let p, s, h be a predicate, a slot and a head word respectively. We then use[6]:

$$AS(s,h) = 1 - Pr(s|h) = 1 - \frac{\Sigma_{p'} N(p', s, h)}{\Sigma_{p',s'} N(p', s', h)}$$

We select the head words of the argument as we did with the selectional preference measure. Again, the AS of the whole argument is defined to be the arithmetic mean of the measure over all of its head words.

Thresholding. In order to turn these measures into classifiers, we set a threshold below which arguments are marked as adjuncts and above which as cores. In order to avoid tuning a parameter for each of the measures, we set the threshold as the median value of this measure in the test set. That is, we find the threshold which tags half of the arguments as cores and half as adjuncts. This relies on the prior knowledge that prepositional arguments are roughly equally divided between cores and adjuncts[7].

3.4 Combination Model

The algorithm proceeds to integrate the predictions of the weak classifiers into a single classifier. We use an ensemble method (Breiman, 1996). Each of the classifiers may either classify an argument as an adjunct, classify it as a core, or abstain. In order to obtain a high accuracy classifier, to be used for self-training below, the ensemble classifier only tags arguments for which none of the classifiers abstained, i.e., when sufficient information was available to make all three predictions. The prediction is determined by the majority vote.

The ensemble classifier has high precision but low coverage. In order to increase its coverage, a self-training step is performed. We observe that a predicate and a slot generally determine whether the argument is a core or an adjunct. For instance, in our development data, a classifier which assigns all arguments that share a predicate and a slot their most common label, yields 94.3% accuracy on the pairs appearing at least 5 times. This property of the core-adjunct distinction greatly simplifies the task for supervised algorithms (see Section 2).

We therefore apply the following procedure: (1) tag the training data with the ensemble classifier; (2) for each test sample x, if more than a ratio of α of the training samples sharing the same predicate and slot with x are labeled as cores, tag x as core. Otherwise, tag x as adjunct.

Test samples which do not share a predicate and a slot with any training sample are considered out of coverage. The parameter α is chosen so half of the arguments are tagged as cores and half as adjuncts. In our experiments α was about 0.25.

4 Experimental Setup

Experiments were conducted in two scenarios. In the '*SID*' (supervised identification of prepositions and verbs) scenario, a gold standard list of prepositions was provided. The list was generated by taking every word tagged by the preposition tag ('*IN*') in at least one of its instances under the gold standard annotation of the WSJ sections 2–21. Verbs were identified using MXPOST (Ratnaparkhi, 1996). Words tagged with any of the verb tags, except of the auxiliary verbs ('have', 'be' and 'do') were considered predicates. This scenario decouples the accuracy of the algorithm from the quality of the unsupervised POS tagging.

In the '*Fully Unsupervised*' scenario, prepositions and verbs were identified using Clark's tagger (Clark, 2003). It was asked to produce a tagging into 34 classes. The classes corresponding to prepositions and to verbs were manually identified. Prepositions in the test set were detected with 84.2% precision and 91.6% recall.

The prediction of whether a word belongs to an open class or a closed was based on the output of the Prototype tagger (Abend et al., 2010). The Prototype tagger provided significantly more ac-

[6] The conditional probability is subtracted from 1 so that higher values correspond to cores, as with the other measures.

[7] In case the test data is small, we can use the median value on the training data instead.

curate predictions in this context than Clark's.

The 39832 sentences of PropBank's sections 2–21 were used as a test set without bounding their lengths[8]. Cores were defined to be any argument bearing the labels 'A0' – 'A5', 'C-A0' – 'C-A5' or 'R-A0' – 'R-A5'. Adjuncts were defined to be arguments bearing the labels 'AM', 'C-AM' or 'R-AM'. Modals ('AM-MOD') and negation modifiers ('AM-NEG') were omitted since they do not represent adjuncts.

The test set includes 213473 arguments, 45939 (21.5%) are prepositional. Of the latter, 22442 (48.9%) are cores and 23497 (51.1%) are adjuncts. The non-prepositional arguments include 145767 (87%) cores and 21767 (13%) adjuncts. The average number of words per argument is 5.1.

The NANC (Graff, 1995) corpus was used as a training set. Only sentences of length not greater than 10 excluding punctuation were used (see Section 3.2), totaling 4955181 sentences. 7673878 (5635810) arguments were identified in the '*SID*' ('*Fully Unsupervised*') scenario. The average number of words per argument is 1.6 (1.7).

Since this is the first work to tackle this task using neither manual nor supervised syntactic annotation, there is no previous work to compare to. However, we do compare against a non-trivial baseline, which closely follows the rationale of cores as obligatory arguments.

Our *Window Baseline* tags a corpus using MXPOST and computes, for each predicate and preposition, the ratio between the number of times that the preposition appeared in a window of W words after the verb and the total number of times that the verb appeared. If this number exceeds a certain threshold β, all arguments having that predicate and preposition are tagged as cores. Otherwise, they are tagged as adjuncts. We used 18.7M sentences from NANC of unbounded length for this baseline. W and β were fine-tuned against the test set[9].

We also report results for partial versions of the algorithm, starting with the three measures used (selectional preference, predicate-slot collocation and argument-slot collocation). Results for the ensemble classifier (prior to the bootstrapping stage) are presented in two variants: one in which the ensemble is used to tag arguments for which all three measures give a prediction (the '*Ensemble(Intersection)*' classifier) and one in which the ensemble tags all arguments for which at least one classifier gives a prediction (the '*Ensemble(Union)*' classifier). For the latter, a tie is broken in favor of the core label. The '*Ensemble(Union)*' classifier is not a part of our model and is evaluated only as a reference.

In order to provide a broader perspective on the task, we compare the measures in the basis of our algorithm to simplified or alternative measures. We experiment with the following measures:

1. *Simple SP* – a selectional preference measure defined to be $Pr(head|slot, predicate)$.

2. *Vast Corpus SP* – similar to '*Simple SP*' but with a much larger corpus. It uses roughly 100M arguments which were extracted from the web-crawling based corpus of (Gabrilovich and Markovitch, 2005) and the British National Corpus (Burnard, 2000).

3. *Thesaurus SP* – a selectional preference measure which follows the paradigm of (Erk, 2007) (Section 3.3) and defines the similarity between two heads to be the Jaccard affinity between their two entries in Lin's automatically compiled thesaurus (Lin, 1998)[10].

4. *Pr(slot|predicate)* – an alternative to the used predicate-slot collocation measure.

5. *PMI(slot, head)* – an alternative to the used argument-slot collocation measure.

6. *Head Dependence* – the entropy of the predicate distribution given the slot and the head (following (Merlo and Esteve Ferrer, 2006)):

$$HD(s,h) = -\Sigma_p Pr(p|s,h) \cdot log(Pr(p|s,h))$$

Low entropy implies a core.

For each of the scenarios and the algorithms, we report accuracy, coverage and effective accuracy. Effective accuracy is defined to be the accuracy obtained when all out of coverage arguments are tagged as adjuncts. This procedure always yields a classifier with 100% coverage and therefore provides an even ground for comparing the algorithms' performance.

We see accuracy as important on its own right since increasing coverage is often straightforward given easily obtainable larger training corpora.

[8]The first 15K arguments were used for the algorithm's development and therefore excluded from the evaluation.

[9]Their optimal value was found to be W=2, β=0.03. The low optimal value of β is an indication of the noisiness of this technique.

[10]Since we aim for a minimally supervised scenario, we used the proximity-based version of his thesaurus which does not require parsing as pre-processing. http://webdocs.cs.ualberta.ca/∼lindek/Downloads/sims.lsp.gz

		Collocation Measures				Ensemble + Cov.	
		Sel. Preference	Pred-Slot	Arg-Slot	Ensemble(I)	Ensemble(U)	E(I) + ST
SID Scenario	Accuracy	65.6	64.5	72.4	**74.1**	68.7	70.6
	Coverage	35.6	77.8	44.7	33.2	**88.1**	74.2
	Eff. Acc.	56.7	64.8	58.8	58.8	67.8	**68.4**
Fully Unsupervised Scenario	Accuracy	62.6	61.1	69.4	**70.6**	64.8	68.8
	Coverage	24.8	59.0	38.7	22.8	**74.2**	56.9
	Eff. Acc.	52.6	57.5	55.8	53.8	61.0	**61.4**

Table 1: Results for the various models. Accuracy, coverage and effective accuracy are presented in percents. Effective accuracy is defined to be the accuracy resulting from labeling each out of coverage argument with an adjunct label. The rows represent the following models (left to right): selectional preference, predicate-slot collocation, argument-slot collocation, '*Ensemble(Intersection)*', '*Ensemble(Union)*' and the '*Ensemble(Intersection)*' followed by self-training (see Section 3.4). '*Ensemble(Intersection)*' obtains the highest accuracy. The ensemble + self-training obtains the highest effective accuracy.

	Selectional Preference Measures				Pred-Slot Measures			Arg-Slot Measures		
	SP*	S. SP	V.C. SP	Lin SP	PS*	Pr(s|p)	Window	AS*	PMI(s, h)	HD
Acc.	**65.6**	41.6	44.8	49.9	**64.5**	58.9	64.1	**72.4**	67.5	67.4
Cov.	35.6	36.9	**45.3**	36.7	77.8	77.8	**92.6**	44.7	44.7	44.7
Eff. Acc.	**56.7**	48.2	47.7	51.3	64.8	60.5	**65.0**	**58.8**	56.6	56.6

Table 2: Comparison of the measures used by our model to alternative measures in the '*SID*' scenario. Results are in percents. The sections of the table are (from left to right): selectional preference measures, predicate-slot measures, argument-slot measures and head dependence. The measures are (left to right): SP*, Simple SP, Vast Corpus SP, Lin SP, PS*, Pr(slot|predicate), Window Baseline, AS*, PMI(slot, head) and Head Dependence. The measures marked with $*$ are the ones used by our model. See Section 4.

Another reason is that a high accuracy classifier may provide training data to be used by subsequent supervised algorithms.

For completeness, we also provide results for the entire set of arguments. The great majority of non-prepositional arguments are cores (87% in the test set). We therefore tag all non-prepositional as cores and tag prepositional arguments using our model. In order to minimize supervision, we distinguish between the prepositional and the non-prepositional arguments using Clark's tagger.

Finally, we experiment on a scenario where even argument identification on the test set is not provided, but performed by the algorithm of (Abend et al., 2009), which uses neither syntactic nor SRL annotation but does utilize a supervised POS tagger. We therefore run it in the '*SID*' scenario. We apply it to the sentences of length at most 10 contained in sections 2–21 of PB (11586 arguments in 6007 sentences). Non-prepositional arguments are invariably tagged as cores and out of coverage prepositional arguments as adjuncts.

We report labeled and unlabeled recall, precision and F-scores for this experiment. An unlabeled match is defined to be an argument that agrees in its boundaries with a gold standard argument and a labeled match requires in addition that the arguments agree in their core/adjunct label. We also report labeling accuracy which is the ratio between the number of labeled matches and the number of unlabeled matches[11].

5 Results

Table 1 presents the results of our main experiments. In both scenarios, the most accurate of the three basic classifiers was the argument-slot collocation classifier. This is an indication that the collocation between the argument and the preposition is more indicative of the core/adjunct label than the obligatoriness of the slot (as expressed by the predicate-slot collocation).

Indeed, we can find examples where adjuncts, although optional, appear very often with a certain verb. An example is 'meet', which often takes a temporal adjunct, as in 'Let's meet [in July]'. This is a semantic property of 'meet', whose syntactic expression is not obligatory.

All measures suffered from a comparable deterioration of accuracy when moving from the '*SID*' to the '*Fully Unsupervised*' scenario. The deterioration in coverage, however, was considerably lower for the argument-slot collocation.

The '*Ensemble(Intersection)*' model in both cases is more accurate than each of the basic classifiers alone. This is to be expected as it combines the predictions of all three. The self-training step significantly increases the ensemble model's cov-

[11] Note that the reported unlabeled scores are slightly lower than those reported in the 2009 paper, due to the exclusion of the modals and negation modifiers.

	Precision	Recall	F-score	lAcc.
Unlabeled	50.7	66.3	57.5	–
Labeled	42.4	55.4	48.0	83.6

Table 3: Unlabeled and labeled scores for the experiments using the unsupervised argument identification system of (Abend et al., 2009). Precision, recall, F-score and labeling accuracy are given in percents.

erage (with some loss in accuracy), thus obtaining the highest effective accuracy. It is also more accurate than the simpler classifier '*Ensemble(Union)*' (although the latter's coverage is higher).

Table 2 presents results for the comparison to simpler or alternative measures. Results indicate that the three measures used by our algorithm (leftmost column in each section) obtain superior results. The only case in which performance is comparable is the window baseline compared to the Pred-Slot measure. However, the baseline's score was obtained by using a much larger corpus and a careful hand-tuning of the parameters[12].

The poor performance of *Simple SP* can be ascribed to sparsity. This is demonstrated by the median value of 0, which this measure obtained on the test set. Accuracy is only somewhat better with a much larger corpus (*Vast Corpus SP*). The *Thesaurus SP* most probably failed due to insufficient coverage, despite its applicability in a similar supervised task (Zapirain et al., 2009).

The Head Dependence measure achieves a relatively high accuracy of 67.4%. We therefore attempted to incorporate it into our model, but failed to achieve a significant improvement to the overall result. We expect a further study of the relations between the measures will suggest better ways of combining their predictions.

The obtained effective accuracy for the entire set of arguments, where the prepositional arguments are automatically identified, was 81.6%.

Table 3 presents results of our experiments with the unsupervised argument identification model of (Abend et al., 2009). The unlabeled scores reflect performance on argument identification alone, while the labeled scores reflect the joint performance of both the 2009 and our algorithms. These results, albeit low, are potentially beneficial for unsupervised subcategorization acquisition. The accuracy of our model on the entire set (prepositional argument subset) of correctly identified arguments was 83.6% (71.7%). This is somewhat higher than the score on the entire test set ('*SID*' scenario), which was 83.0% (68.4%), probably due to the bounded length of the test sentences in this case.

[12]We tried about 150 parameter pairs for the baseline. The average of the five best effective accuracies was 64.3%.

6 Conclusion

We presented a fully unsupervised algorithm for the classification of arguments into cores and adjuncts. Since most non-prepositional arguments are cores, we focused on prepositional arguments, which are roughly equally divided between cores and adjuncts. The algorithm computes three statistical measures and utilizes ensemble-based and self-training methods to combine their predictions.

The algorithm applies state-of-the-art unsupervised parser and POS tagger to collect statistics from a large raw text corpus. It obtains an accuracy of roughly 70%. We also show that (somewhat surprisingly) an argument-slot collocation measure gives more accurate predictions than a predicate-slot collocation measure on this task. We speculate the reason is that the head word disambiguates the preposition and that this disambiguation generally determines whether a prepositional argument is a core or an adjunct (somewhat independently of the predicate). This calls for a future study into the semantics of prepositions and their relation to the core-adjunct distinction. In this context two recent projects, *The Preposition Project* (Litkowski and Hargraves, 2005) and *PrepNet* (Saint-Dizier, 2006), which attempt to characterize and categorize the complex syntactic and semantic behavior of prepositions, may be of relevance.

It is our hope that this work will provide a better understanding of core-adjunct phenomena. Current supervised SRL models tend to perform worse on adjuncts than on cores (Pradhan et al., 2008; Toutanova et al., 2008). We believe a better understanding of the differences between cores and adjuncts may contribute to the development of better SRL techniques, in both its supervised and unsupervised variants.

References

Omri Abend, Roi Reichart and Ari Rappoport, 2009. *Unsupervised Argument Identification for Semantic Role Labeling.* ACL '09.

Omri Abend, Roi Reichart and Ari Rappoport, 2010. *Improved Unsupervised POS Induction through Prototype Discovery.* ACL '10.

Collin F. Baker, Charles J. Fillmore and John B. Lowe, 1998. *The Berkeley FrameNet Project.* ACL-COLING '98.

Timothy Baldwin, Valia Kordoni and Aline Villavicencio, 2009. *Prepositions in Applications: A Survey and Introduction to the Special Issue.* Computational Linguistics, 35(2):119–147.

Ram Boukobza and Ari Rappoport, 2009. *Multi-Word Expression Identification Using Sentence Surface Features.* EMNLP '09.

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

Ted Briscoe and John Carroll, 1997. *Automatic Extraction of Subcategorization from Corpora.* Applied NLP '97.

Lou Burnard, 2000. *User Reference Guide for the British National Corpus.* Technical report, Oxford University.

Xavier Carreras and Lluìs Màrquez, 2005. *Introduction to the CoNLL–2005 Shared Task: Semantic Role Labeling.* CoNLL '05.

Alexander Clark, 2003. *Combining Distributional and Morphological Information for Part of Speech Induction.* EACL '03.

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

David Dowty, 2000. *The Dual Analysis of Adjuncts and Complements in Categorial Grammar.* Modifying Adjuncts, ed. Lang, Maienborn and Fabricius–Hansen, de Gruyter, 2003.

Katrin Erk, 2007. *A Simple, Similarity-based Model for Selectional Preferences.* ACL '07.

Evgeniy Gabrilovich and Shaul Markovitch, 2005. *Feature Generation for Text Categorization using World Knowledge.* IJCAI '05.

David Graff, 1995. *North American News Text Corpus.* Linguistic Data Consortium. LDC95T21.

Trond Grenager and Christopher D. Manning, 2006. *Unsupervised Discovery of a Statistical Verb Lexicon.* EMNLP '06.

Donald Hindle and Mats Rooth, 1993. *Structural Ambiguity and Lexical Relations.* Computational Linguistics, 19(1):103–120.

Julia Hockenmaier, 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar.* Ph.D. thesis, University of Edinburgh.

Karin Kipper, Hoa Trang Dang and Martha Palmer, 2000. *Class-Based Construction of a Verb Lexicon.* AAAI '00.

Anna Korhonen, 2002. *Subcategorization Acquisition.* Ph.D. thesis, University of Cambridge.

Hang Li and Naoki Abe, 1998. *Generalizing Case Frames using a Thesaurus and the MDL Principle.* Computational Linguistics, 24(2):217–244.

Wei Li, Xiuhong Zhang, Cheng Niu, Yuankai Jiang and Rohini Srihari, 2003. *An Expert Lexicon Approach to Identifying English Phrasal Verbs.* ACL '03.

Dekang Lin, 1998. *Automatic Retrieval and Clustering of Similar Words.* COLING–ACL '98.

Ken Litkowski and Orin Hargraves, 2005. *The Preposition Project.* ACL-SIGSEM Workshop on "The Linguistic Dimensions of Prepositions and Their Use in Computational Linguistic Formalisms and Applications".

Diana McCarthy, 2001. *Lexical Acquisition at the Syntax-Semantics Interface: Diathesis Alternations, Subcategorization Frames and Selectional Preferences.* Ph.D. thesis, University of Sussex.

Paula Merlo and Eva Esteve Ferrer, 2006. *The Notion of Argument in Prepositional Phrase Attachment.* Computational Linguistics, 32(3):341–377.

Martha Palmer, Daniel Gildea and Paul Kingsbury, 2005. *The Proposition Bank: A Corpus Annotated with Semantic Roles.* Computational Linguistics, 31(1):71–106.

Sameer Pradhan, Wayne Ward and James H. Martin, 2008. *Towards Robust Semantic Role Labeling.* Computational Linguistics, 34(2):289–310.

Vasin Punyakanok, Dan Roth and Wen-tau Yih, 2008. *The Importance of Syntactic Parsing and Inference in Semantic Role Labeling.* Computational Linguistics, 34(2):257–287.

Adwait Ratnaparkhi, 1996. *Maximum Entropy Part-Of-Speech Tagger.* EMNLP '96.

Roi Reichart, Omri Abend and Ari Rappoport, 2010. *Type Level Clustering Evaluation: New Measures and a POS Induction Case Study.* CoNLL '10.

Philip Resnik, 1996. *Selectional constraints: An information-theoretic model and its computational realization.* Cognition, 61:127–159.

Patrick Saint-Dizier, 2006. *PrepNet: A Multilingual Lexical Description of Prepositions.* LREC '06.

Anoop Sarkar and Daniel Zeman, 2000. *Automatic Extraction of Subcategorization Frames for Czech.* COLING '00.

Sabine Schulte im Walde, Christian Hying, Christian Scheible and Helmut Schmid, 2008. *Combining EM Training and the MDL Principle for an Automatic Verb Classification Incorporating Selectional Preferences.* ACL '08.

Yoav Seginer, 2007. *Fast Unsupervised Incremental Parsing.* ACL '07.

Caroline Sporleder and Linlin Li, 2009. *Unsupervised Recognition of Literal and Non-Literal Use of Idiomatic Expressions.* EACL '09.

Robert S. Swier and Suzanne Stevenson, 2004. *Unsupervised Semantic Role Labeling.* EMNLP '04.

Robert S. Swier and Suzanne Stevenson, 2005. *Exploiting a Verb Lexicon in Automatic Semantic Role Labelling.* EMNLP '05.

Kristina Toutanova, Aria Haghighi and Christopher D. Manning, 2008. *A Global Joint Model for Semantic Role Labeling.* Computational Linguistics, 34(2):161–191.

Aline Villavicencio, 2002. *Learning to Distinguish PP Arguments from Adjuncts.* CoNLL '02.

Dave Willis, 2004. *Collins Cobuild Intermedia English Grammar, Second Edition*. HarperCollins Publishers.

Nianwen Xue and Martha Palmer, 2004. *Calibrating Features for Semantic Role Labeling.* EMNLP '04.

Beñat Zapirain, Eneko Agirre and Lluís Màrquez, 2009. *Generalizing over Lexical Features: Selectional Preferences for Semantic Role Classification.* ACL '09, short paper.