A Knowledge-Intensive Model for Prepositional Phrase Attachment

Ndapandula Nakashole
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15213
ndapa@cs.cmu.edu

Tom M. Mitchell
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15213
tom.mitchell@cs.cmu.edu

Abstract

Prepositional phrases (PPs) express crucial information that knowledge base construction methods need to extract. However, PPs are a major source of syntactic ambiguity and still pose problems in parsing. We present a method for resolving ambiguities arising from PPs, making extensive use of semantic knowledge from various resources. As training data, we use both labeled and unlabeled data, utilizing an expectation maximization algorithm for parameter estimation. Experiments show that our method yields improvements over existing methods including a state of the art dependency parser.

1 Introduction

Machine reading and information extraction (IE) projects have produced large resources with many millions of facts (Suchanek et al., 2007; Mitchell et al., 2015). This wealth of knowledge creates a positive feedback loop for automatic knowledge base construction efforts: the accumulated knowledge can be leveraged to improve machine reading; in turn, improved reading methods can be used to better extract knowledge expressed using complex and potentially ambiguous language. For example, prepositional phrases (PPs) express crucial information that IE methods need to extract. However, PPs are a major source of syntactic ambiguity. In this paper, we propose to use semantic knowledge to improve PP attachment disambiguation. PPs such as "in", "at", and "for" express details about the *where, when,* and *why* of relations and events. PPs also state attributes of nouns.

As an example, consider the following sentences: *S1.) Alice caught the butterfly with the spots. S2.) Alice caught the butterfly with the net.*

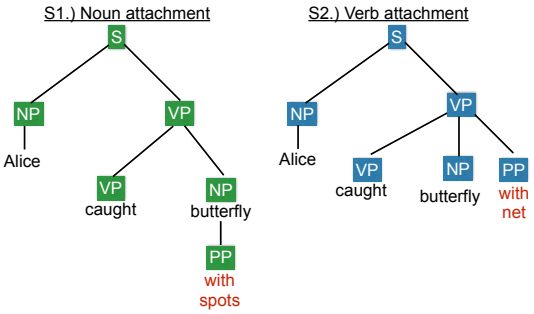

Figure 1: Parse trees where the prepositional phrase (PP) attaches to the noun, and to the verb.

Relations	Noun-Noun binary relations *(Paris, located in, France)* *(net, caught, butterfly)*
Nouns	Noun semantic categories *(butterfly, isA, animal)*
Verbs	Verb roles *caught(agent, patient, instrument)*
Prepositions	Preposition definitions *f(for)= used for, has purpose, ...* *f(with)= has, contains, ...*
Discourse	Context $n0 \in \{n0, v, n1, p, n2\}$

Table 1: Types of background knowledge used in this paper to determine PP attachment.

S1 and S2 are syntactically different, this is evident from their corresponding parse trees in Figure 1. Specifically, S1 and S2 differ in where their PPs attach. In S1, the butterfly has spots and therefore the PP, "with the spots", attaches to the *noun*. For relation extraction, we obtain a *binary* relation of the form: ⟨Alice⟩ caught ⟨butterfly with spots⟩. However, in S2, the net is the instrument used for catching and therefore the PP, "with the net", attaches to the *verb*. For relation extraction, we get a *ternary* extraction of the form: ⟨Alice⟩ caught ⟨butterfly⟩ with ⟨net⟩.

The PP attachment problem is often defined as follows: given a PP occurring within a sentence where there are multiple possible attachment sites

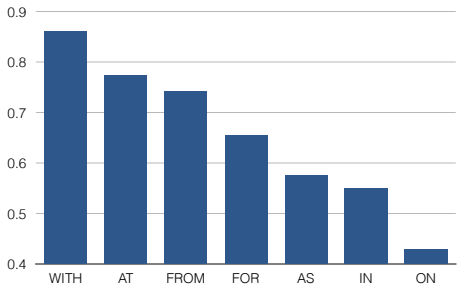

Figure 2: Dependency parser PP attachment accuracy for various frequent prepositions.

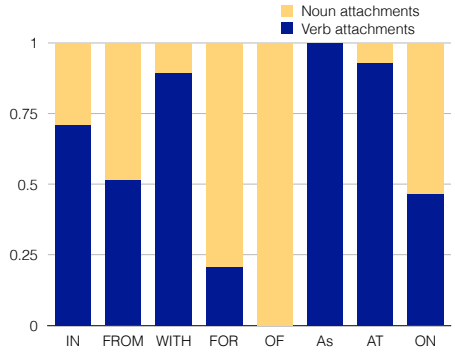

Figure 3: Noun vs. verb attachment proportions for frequent prepositions in the labeled NYTC dataset.

for the PP, choose the most plausible attachment site. In the literature, prior work going as far back as (Brill and Resnik, 1994; Ratnaparkhi et al., 1994; Collins and Brooks, 1995) has focused on the language pattern that causes most PP ambiguities, which is the 4-word sequence: $\{v, n1, p, n2\}$ (e.g., $\{$*caught, butterfly, with, spots*$\}$). The task is to determine if the prepositional phrase $(p, n2)$ attaches to the verb v or to the first noun $n1$. Following common practice, we focus on PPs occurring as $\{v, n1, p, n2\}$ quadruples — we shall refer to these as *PP quads*.

The approach we present here differs from prior work in two main ways. First, we make extensive use of semantic knowledge about nouns, verbs, prepositions, pairs of nouns, and the discourse context in which a PP quad occurs. Table 1 summarizes the types of knowledge we considered in our work. Second, in training our model, we rely on both labeled and unlabeled data, employing an expectation maximization (EM) algorithm (Dempster et al., 1977).

Contributions. In summary, our main contributions are:

1) Semantic Knowledge: Previous methods largely rely on corpus statistics. Our approach draws upon diverse sources of background knowledge, leading to performance improvements.

2) Unlabeled Data: In addition to training on labeled data, we also make use of a large amount of unlabeled data. This enhances our method's ability to generalize to diverse data sets.

3) Datasets: In addition to the standard Wall Street Journal corpus (WSJ) (Ratnaparkhi et al., 1994), we labeled two new datasets for testing purposes, one from Wikipedia (WKP), and another from the New York Times Corpus (NYTC). We make these datasets freely available for future research. In addition, we have applied our model to over 4 million 5-tuples of the form $\{n0, v, n1, p, n2\}$, and we also make this dataset available[1] for research into ternary relation extraction beyond spatial and temporal scoping.

2 State of the Art

To quantitatively assess existing tools, we analyzed performance of the widely used Stanford parser[2] as of 2014, and the established baseline algorithm (Collins and Brooks, 1995), which has stood the test of time. We first manually labeled PP quads from the NYTC dataset, then prepended the noun phrase appearing before the quad, effectively creating sentences made up of 5 lexical items $(n0\ v\ n1\ p\ n2)$. We then applied the Stanford parser, obtaining the results summarized in Figure 2. The parser performs well on some prepositions, for example, "of", which tends to occur with noun attaching PPs as can be seen in Figure 3. However, for prepositions with an even distribution over verb and noun attachments, such as "on", precision is as low as 50%. The Collins baseline achieves 84% accuracy on the benchmark Wall Street Journal PP dataset. However, drawing a distinction in the precision of different prepositions provides useful insights on its performance. We re-implemented this baseline and found that when we remove the trivial preposition, "of", whose PPs are by default attached to the noun by this baseline, precision drops to 78%. This analysis suggests there is substantial room for improvement.

[1] http://rtw.ml.cmu.edu/resources/ppa
[2] http://nlp.stanford.edu:8080/parser/

3 Related Work

Statistics-based Methods. Prominent prior methods learn to perform PP attachment based on corpus co-occurrence statistics, gathered either from manually annotated training data (Collins and Brooks, 1995; Brill and Resnik, 1994) or from automatically acquired training data that may be noisy (Ratnaparkhi, 1998; Pantel and Lin, 2000). These models collect statistics on how often a given quadruple, $\{v, n1, p, n2\}$, occurs in the training data as a verb attachment as opposed to a noun attachment. The issue with this approach is sparsity, that is, many quadruples occuring in the test data might not have been seen in the training data. Smoothing techniques are often employed to overcome sparsity. For example, (Collins and Brooks, 1995) proposed a back-off model that uses subsets of the words in the quadruple, by also keeping frequency counts of triples, pairs and single words. Another approach to overcoming sparsity has been to use WordNet (Fellbaum, 1998) classes, by replacing nouns with their WordNet classes (Stetina and Nagao, 1997; Toutanova et al., 2004) to obtain less sparse corpus statistics. Corpus-derived clusters of similar nouns and verbs have also been used (Pantel and Lin, 2000).

Hindle and Rooth proposed a lexical association approach based on how words are associated with each other (Hindle and Rooth, 1993). Lexical preference is used by computing co-occurrence frequencies (lexical associations) of verbs and nouns, with prepositions. In this manner, they would discover that, for example, the verb "send" is highly associated with the preposition *from*, indicating that in this case, the PP is likely to be a verb attachment.

Structure-based Methods. These methods are based on high-level observations that are then generalized into heuristics for PP attachment decisions. (Kimball, 1988) proposed a right association method, whose premise is that a word tends to attach to another word immediately to its right. (Frazier, 1978) introduced a minimal attachment method, which posits that words attach to an existing non-terminal word using the fewest additional syntactic nodes. While simple, in practice these methods have been found to perform poorly (Whittemore et al., 1990).

Rule-based Methods. (Brill and Resnik, 1994) proposed methods that learn a set of transformation rules from a corpus. The rules can be too specific to have broad applicability, resulting in low recall. To address low recall, knowledge about nouns, as found in WordNet, is used to replace certain words in rules with their WordNet classes.

Parser Correction Methods. The quadruples formulation of the PP problem can be seen as a simplified setting. This is because, with quadruples, there is no need to deal with complex sentences but only well-defined quadruples of the form $\{v, n1, p, n2\}$. Thus in the quadruples setting, there are only two possible attachment sites for the PP, the v and $n1$. An alternative setting is to work in the context of full sentences. In this setting the problem is cast as a dependency parser correction problem (Atterer and Schütze, 2007; Agirre et al., 2008; Anguiano and Candito, 2011). That is, given a dependency parse of a sentence, with potentially incorrect PP attachments, rectify it such that the prepositional phrases attach to the correct sites. Unlike our approach, these methods do not take semantic knowledge into account.

Sense Disambiguation. In addition to prior work on prepositional phrase attachment, a highly related problem is preposition sense disambiguation (Hovy et al., 2011; Srikumar and Roth, 2013). Even a syntactically correctly attached PP can still be semantically ambiguous with respect to questions of machine reading such as *where, when,* and *why*. Therefore, when extracting information from prepositions, the problem of preposition sense disambiguation (semantics) has to be addressed in addition to prepositional phrase attachment disambiguation (syntax). In this paper, our focus is on the latter.

4 Methodology

Our approach consists of first generating features from background knowledge and then training a model to learn with these features. The types of features considered in our experiments are summarized in Table 2. The choice of features was motivated by our empirically driven characterization of the problem as follows:

(Verb attach) $\longrightarrow$ *v* $\langle$*has-slot-filler*$\rangle$ *n2*
(Noun attach a.) $\longrightarrow$ *n1* $\langle$*described-by*$\rangle$ *n2*
(Noun attach b.) $\longrightarrow$ *n2* $\langle$*described-by*$\rangle$ *n1*

Feature Type	#	Feature	Example
Noun-Noun Binary Relations		**Source: SVOs**	
	F1.	$svo(n2, v, n1)$	For q1; $(net, caught, butterfly)$
	F2.	$\forall i : \exists sv_io$; $svo(n1, v_i, n2)$	For q2; $(butterfly, has, spots)$
			For q2; $(butterfly, can\ see, spots)$
Noun Semantic Categories		**Source:** $\mathcal{T}$	
	F3.	$\forall t_i \in \mathcal{T}; isA(n1, t_i)$	For q1 $isA(butterlfy, animal)$
	F4.	$\forall t_i \in \mathcal{T}; isA(n2, t_i)$	For q2 $isA(net, device)$
Verb Role Fillers		**Source: VerbNet**	
	F5.	$hasRole(n2, r_i)$	For q1; $(net, instrument)$
Preposition Relational		**Source:** $\mathcal{M}$	
Definitions	F6.	$def(prep, v_i)\ \forall i :$	
		$\exists sv_io; v_i \in \mathcal{M} \wedge$	
		$svo(n1, v_i, n2)$	For q2; $def(with, has)$
Discourse Features		**Source: Sentence(s),** $\mathcal{T}$	
	F7.	$\forall t_i \in \mathcal{T}; isA(n0, t_i)$	$n0 \in \{n0, v, n1, p, n2\}$
Lexical Features		**Source: PP quads**	For q1;
	F8.	$(v, n1, p, n2)$	$(caught, butterfly, with, net)$
	F9.	$(v, n1, p)$	$(caught, butterfly, with)$
	F10.	$(v, p, n2)$	$(caught, with, net)$
	F11.	$(n1, p, n2)$	$(butterfly, with, net)$
	F12.	(v, p)	$(caught, with)$
	F13.	$(n1, p)$	$(butterfly, with)$
	F14.	$(p, n2)$	$(with, net)$
	F15.	(p)	$(with)$

Table 2: Types of features considered in our experiments. All features have values of 1 or 0. The PP quads used as running examples are: $q1 = \{caught, butterfly, with, net\} : V$, $q2 = \{caught, butterfly, with, spots\} : N$.

That is, we found that for verb-attaching PPs, $n2$ is usually a role filler for the verb, e.g., the net fills the role of an instrument for the verb $catch$. On the other hand, for noun-attaching PPs, one noun describes or elaborates on the other. In particular, we found two kinds of noun attachments. For the first kind of noun attachment, the second noun $n2$ describes the first noun $n1$, for example $n2$ might be an attribute or property of $n1$, as in the spots($n2$) are an attribute of the butterfly ($n1$). And for the second kind of noun attachment, the first noun $n1$ describes the second noun $n2$, as in the PP quad {*expect, decline, in, rates*}, where the PP "in rates", attaches to the $noun$. The decline:$n1$ that is expected:v is in the rates:$n2$. We sampled 50 PP quads from the WSJ dataset and found that every labeling could be explained using our characterization. We make this labeling available with the rest of the datasets.

We next describe in more detail how each type of feature is derived from the background knowledge in Table 1.

4.1 Feature Generation

We generate boolean-valued features for all the feature types we describe in this section.

4.1.1 Noun-Noun Binary Relations

The noun-noun binary relation features, F1-2 in Table 2, are boolean features $svo(n1, v_i, n2)$ (where v_i is any verb) and $svo(n2, v, n1)$ (where v is the verb in the PP quad, and the roles of $n2$ and $n1$ are reversed). These features describe diverse semantic relations between pairs of nouns (e.g., *butterfly-has-spots*, *clapton-played-guitar*). To obtain this type of knowledge, we dependency parsed all sentences in the 500 million English web pages of the ClueWeb09 corpus, then extracted subject-verb-object (SVO) triples from these parses, along with the frequency of

each SVO triple in the corpus. The value of any given feature $svo(n1, v_i, n2)$ is defined to be 1 if that SVO triple was found at least 3 times in these SVO triples, and 0 otherwise. To see why these relations are relevant, let us suppose that we have the knowledge that *butterfly-has-spots*, $svo(n1, v_i, n2)$. From this, we can infer that the PP in $\{caught, butterfly, with, spots\}$ is likely to attach to the noun. Similarly, suppose we know that *net-caught-butterfly*, $svo(n2, v, n1)$. The fact that a net can be used to catch a butterfly can be used to predict that the PP in $\{caught, butterfly, with, net\}$ is likely to attach to the verb.

4.1.2 Noun Semantic Categories

Noun semantic type features, F3-4, are boolean features $isA(n1, t_i)$ and $isA(n2, t_i)$ where t_i is a noun category in a noun categorization scheme $\mathcal{T}$ such as WordNet classes. Knowledge about semantic types of nouns, for example that a butterfly is an animal, enables extrapolating predictions to other PP quads that contain nouns of the same type. We ran experiments with several noun categorizations including WordNet classes, knowledge base ontological types, and an unsupervised noun categorization produced by clustering nouns based on the verbs and adjectives with which they co-occur (distributional similarity).

4.1.3 Verb Role Fillers

The verb role feature, F5, is a boolean feature $hasRole(n2, r_i)$ where r_i is a role that $n2$ can fulfill for the verb v in the PP quad, according to background knowledge. Notice that if $n2$ fills a role for the verb, then the PP is a verb attachment. Consider the quad $\{caught, butterfly, with, net\}$, if we know that a net can play the role of an *instrument* for the verb *catch*, this suggests a likely verb attachment. We obtained background knowledge of verbs and their possible roles from the VerbNet lexical resource (Kipper et al., 2008). From VerbNet we obtained $2,573$ labeled sentences containing PP quads (verbs in the same VerbNet group are considered synonymous), and the labeled semantic roles filled by the second noun $n2$ in the PP quad. We use these example sentences to label similar PP quads, where similarity of PP quads is defined by verbs from the same VerbNet group.

4.1.4 Preposition Definitions

The preposition definition feature, $F6$, is a boolean feature $def(prep, v_i) = 1 \; if \; \exists v_i \in \mathcal{M} \wedge svo(n1, v_i, n2) = 1$, where $\mathcal{M}$ is a definition mapping of prepositions to verb phrases. This mapping defines prepositions, using verbs in our ClueWeb09 derived SVO corpus, in order to capture their senses using verbs; it contains definitions such as *def(with, *) = contains, accompanied by, ...* . If "with" is used in the sense of "contains" , then the PP is a likely noun attachment, as in $n1$ contains $n2$ in the quad $ate, cookies, with, cranberries$. However, if "with" is used in the sense of "accompanied by", then the PP is a likely verb attachment, as in the quad $visted, Paris, with, Sue$. To obtain the mapping, we took the labeled PP quads (WSJ, (Ratnaparkhi et al., 1994)) and computed a ranked list of verbs from SVOs, that appear frequently between pairs of nouns for a given preposition. Other sample mappings are: *def(for,*)= used for*, *def(in,*)= located in*. Notice that this feature $F6$ is a selective, more targeted version of $F2$.

4.1.5 Discourse and Lexical Features

The discourse feature, $F7$, is a boolean feature $isA(n0, t_i)$, for each noun category t_i found in a noun category ontology $\mathcal{T}$ such as WordNet semantic types. The context of the PP quad can contain relevant information for attachment decisions. We take into account the noun preceding a PP quad, in particular, its semantic type. This in effect makes the PP quad into a PP 5-tuple: $\{n0, v, n1, p, n2\}$, where the $n0$ provides additional context.

Finally, we use lexical features in the form of PP quads, features F8-15. To overcome sparsity of occurrences of PP quads, we also use counts of shorter sub-sequences, including triples, pairs and singles. We only use sub-sequences that contain the preposition, as the preposition has been found to be highly crucial in PP attachment decisions (Collins and Brooks, 1995).

4.2 Disambiguation Algorithm

We use the described features to train a model for making PP attachment decisions. Our goal is to compute $\mathbb{P}(y|x)$, the probability that the PP $(p, n2)$ in the tuple $\{v, n1, p, n2\}$ attaches to the *verb (v)* , $y = 1$ or to the $noun(n1)$, $y = 0$, given

a feature vector x describing that tuple. As input to training the model, we are given a collection of PP quads, D where $d_i \in \mathcal{D} : d_i = \{v, n1, p, n2\}$. A small subset, $D^l \subset \mathcal{D}$ is labeled data, thus for each $d_i \in D^l$ we know the corresponding y_i. The rest of the quads, D^u, are unlabeled, hence their corresponding y_is are unknown. From each PP quad d_i, we extract a feature vector x_i according to the feature generation process discussed in Section 4.1.

4.2.1 Model

To model $\mathbb{P}(y|x)$, there a various possibilities. One could use a generative model (e.g., Naive Bayes) or a discriminative model (e.g., logistic regression). In our experiments we used both kinds of models, but found the discriminative model performed better. Therefore, we present details only for our discriminative model. We use the logistic function: $\mathbb{P}(y|x, \vec{\theta}) = \frac{e^{\vec{\theta}x}}{1+e^{\vec{\theta}x}}$, where $\vec{\theta}$ is a vector of model parameters. To estimate these parameters, we could use the labeled data as training data and use standard gradient descent to minimize the logistic regression cost function. However, we also leverage the unlabeled data.

4.2.2 Parameter Estimation

To estimate model parameters based on both labeled and unlabeled data, we use an Expectation Maximization (EM) algorithm. EM estimates model parameters that maximize the expected log likelihood of the full (observed and unobserved) data. Since we are using a discriminative model, our likelihood function is a conditional likelihood function:

$$\begin{aligned}\mathcal{L}(\theta) &= \sum_{i=1}^{N} \ln \mathbb{P}(y_i|x_i) \\ &= \sum_{i=1}^{N} y_i \theta^T x_i - \ln\left(1 + exp(\theta^T x_i)\right) \quad (1)\end{aligned}$$

where i indexes over the N training examples.

The EM algorithm produces parameter estimates that correspond to a local maximum in the expected log likelihood of the data under the posterior distribution of the labels, given by: $\arg\max_{\theta} E_{p(y|x,\theta)}[\ln \mathbb{P}(y|x, \theta)]$. In the E-step, we use the current parameters θ^{t-1} to compute the posterior distribution over the y labels, give by $\mathbb{P}(y|x, \theta^{t-1})$. We then use this posterior distribution to find the expectation of the log of the complete-data conditional likelihood, this expectation is given by $\mathcal{Q}(\theta, \theta^{\mathbf{t-1}})$, defined as:

$$\mathcal{Q}(\theta, \theta^{t-1}) = \sum_{i=1}^{N} E_{\theta^{t-1}}[\ln \mathbb{P}(y|x, \theta)] \quad (2)$$

In the M-step, a new estimate θ^t is then produced, by maximizing this Q function with respect to θ:

$$\theta^{\mathbf{t}} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta^{\mathbf{t-1}}) \quad (3)$$

EM iteratively computes parameters $\theta^0, \theta^1, ...\theta^t$, using the above update rule at each iteration t, halting when there is no further improvement in the value of the Q function. Our algorithm is summarized in Algorithm 1. The M-step solution for θ^t is obtained using gradient ascent to maximize the Q function.

Algorithm 1 The EM algorithm for PP attachment

Input: $\mathcal{X}, \mathcal{D} = D^l \cup D^u$
Output: θ^T
for t = 1 . . . T **do**
  E-Step:
  Compute $p(y|x_i, \theta^{t-1})$
  $x_i : d_i \in D^u; p(y|x_i, \vec{\theta}) = \frac{e^{\vec{\theta}x}}{1+e^{\vec{\theta}x}}$
  $x_i : d_i \in D^l; p(y|x_i) = 1$ if $y = y_i$, else 0
  M-Step:
  Compute new parameters, θ^t
  $\theta^{\mathbf{t}} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta^{\mathbf{t-1}})$

$$\mathcal{Q}(\theta, \theta^{t-1}) = \sum_{i=1}^{N} \sum_{y \in \{0,1\}} p(y|x_i, \theta^{t-1}) \times (y\theta^T x_i - \ln(1 + exp(\theta^T x_i)))$$

  if convergence($\mathcal{L}(\theta), \mathcal{L}(\theta^{t-1})$) **then**
    break
  end if
end for
return θ^T

5 Experimental Evaluation

We evaluated our method on several datasets containing PP quads of the form $\{v, n1, p, n2\}$. The task is to predict if the PP $(p, n2)$ attaches to the verb v or to the first noun $n1$.

5.1 Experimental Setup

Datasets. Table 3 shows the datasets used in our experiments. As labeled training data, we used the

DataSet	# Training quads	# Test quads
Labeled data		
WSJ	20,801	3,097
NYTC	0	293
WKP	0	381
Unlabeled data		
WKP	100,000	4,473,072

Table 3: Training and test datasets used in our experiments.

	PPAD	PPAD-NB	Collins	Stanford
WKP	**0.793**	0.740	0.727	0.701
WKP \of	**0.759**	0.698	0.683	0.652
NYTC	**0.843**	0.792	0.809	0.679
NYTC \of	**0.815**	0.754	0.774	0.621
WSJ	**0.843**	0.816	0.841	N\A
WSJ \of	**0.779**	0.741	0.778	N\A

Table 4: PPAD vs. baselines.

Wall Street Journal (WSJ) dataset. For the unlabeled training data, we extracted PP quads from Wikipedia (WKP) and randomly selected $100,000$ which we found to be a sufficient amount of unlabeled data. The largest labeled test dataset is WSJ but it is also made up of a large fraction, of "of" PP quads, 30% , which trivially attach to the noun, as already seen in Figure 3. The New York Times (NYTC) and Wikipedia (WKP) datasets are smaller but contain fewer proportions of "of" PP quads, 15%, and 14%, respectively. Additionally, we applied our model to over 4 million unlabeled 5-tuples from Wikipedia. We make this data available for download, along with our manually labeled NYTC and WKP datasets. For the WKP & NYTC corpora, each quad has a preceding noun, $n0$, as context, resulting in PP 5-tuples of the form: $\{n0, v, n1, p, n2\}$. The WSJ dataset was only available to us in the form of PP quads with no other sentence information.

Methods Under Comparison. *1) PPAD* (Prepositional Phrase Attachment Disambiguator) is our proposed method. It uses diverse types of semantic knowledge, a mixture of labeled and unlabeled data for training data, a logistic regression classifier, and expectation maximization (EM) for parameter estimation *2) Collins* is the established baseline among PP attachment algorithms (Collins and Brooks, 1995). *3) Stanford Parser* is a state-of-the-art dependency parser, the 2014 online version. *4) PPAD Naive Bayes(NB)* is the same as PPAD but uses a generative model, as opposed to the discriminative model used in PPAD.

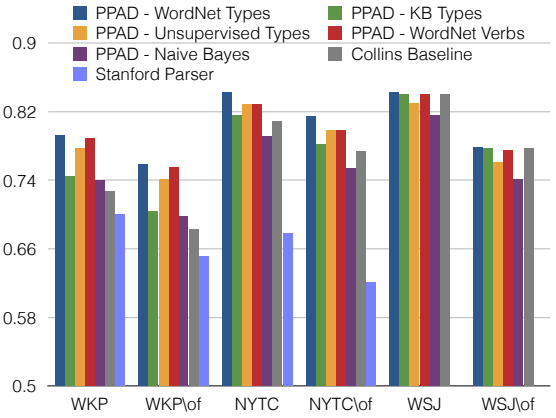

Figure 4: PPAD variations vs. baselines.

5.2 PPAD vs. Baselines

Comparison results of our method to the three baselines are shown in Table 4. For each dataset, we also show results when the "of" quads are removed, shown as "WKP\of", "NYTC\of", and "WSJ\of". Our method yields improvements over the baselines. Improvements are especially significant on the datasets for which no labeled data was available (NYTC and WKP). On WKP, our method is 7% and 9% ahead of the Collins baseline and the Stanford parser, respectively. On NYTC, our method is 4% and 6% ahead of the Collins baseline and the Stanford parser, respectively. On WSJ, which is the source of the labeled data, our method is not significantly better than the Collins baseline. We could not evaluate the Stanford parser on the WSJ dataset. The parser requires well-formed sentences which we could not generate from the WSJ dataset as it was only available to us in the form of PP quads with no other sentence information. For the same reason, we could not generate discourse features,$F7$, for the WSJ PP quads. For the NYTC and WKP datasets, we generated well-formed short sentences containing only the PP quad and the noun preceding it.

Feature Type	Precision	Recall	F1
Noun-Noun Binary Relations (F1-2)	low	$high$	low
Noun Semantic Categories (F3-4)	$high$	$high$	**high**
Verb Role Fillers (F5)	$high$	low	low
Preposition Definitions (F6)	low	low	low
Discourse Features (F7)	$high$	low	**high**
Lexical Features (F8-15)	$high$	$high$	**high**

Table 5: An approximate characterization of feature knowledge sources in terms of precision/recall/F1

5.3 Feature Analysis

We found that features $F2$ and $F6$ did not improve performance, therefore we excluded them from the final model, PPAD. This means that binary noun-noun relations were not useful when used permissively, feature $F2$, but when used selectively, feature $F1$, we found them to be useful. Our attempt at mapping prepositions to verb definitions produced some noisy mappings, resulting in feature $F6$ producing mixed results. To analyze the impact of the unlabeled data, we inspected the features and their weights as produced by the PPAD model. From the unlabeled data, new lexical features were discovered that were not in the original labeled data. Some sample new features with high weights for verb attachments are: *(perform,song,for,*), (lose,*,by,*), (buy,property,in,*).* And for noun attachments: *(*,conference,on,*), (obtain,degree,in,*), (abolish,taxes,on,*).*

We evaluated several variations of PPAD, the results are shown in Figure 4. For "PPAD-WordNet Verbs", we expanded the data by replacing verbs in PP quads with synonymous WordNet verbs, ignoring verb senses. This resulted in more instances of features F1, F8-10, & F12.

We also used different types of noun categorizations: WordNet classes, semantic types from the NELL knowledge base (Mitchell et al., 2015) and unsupervised types. The KB types and the unsupervised types did not perform well, possibly due to the noise found in these categorizations. WordNet classes showed the best results, hence they were used in the final PPAD model for features F3-4 & F7. In Section 5.1, PPAD corresponds to the best model.

5.4 Discussion: The F1 Score of Knowledge

Why did we not reach 100% accuracy? Should relational knowledge not be providing a much bigger performance boost than we have seen in the results? To answer these questions, we characterize our features in terms precision and recall, and F1 measure of their knowledge sources in Table 5. A low recall feature means that the feature does not fire on many examples, the feature's knowledge source suffers from low coverage. A low precision feature means that when it fires, the feature could be incorrect, the feature's knowledge source contains a lot of errors.

From Table 5, the noun-noun binary relation features $(F1-2)$ have low precision, but high recall. This is because the SVO data, extracted from the ClueWeb09 corpus, that we used as our relational knowledge source is very noisy but it is high coverage. The low precision of the SVO data causes these features to be detrimental to performance. Notice that when we used a filtered version of the data, in feature $F2$, the data was no longer detrimental to performance. However, the $F2$ feature is low recall, and therefore it's impact on performance is also limited. The noun semantic category features $(F3-4)$ have high recall and precision, hence it to be expected that their impact on performance is significant. The verb role filler features $(F5)$, obtained from VerbNet have high precision but low recall, hence their marginal impact on performance is also to be expected. The preposition definition features $(F6)$ poor precision made them unusable. The discourse features $(F7)$ are based noun semantic types and lexical features $(F8-15)$, both of which have high recall and precision, hence they useful impact on performance.

In summary, low precision in knowledge is detrimental to performance. In order for knowledge to make even more significant contributions to language understanding, high precision, high recall knowledge sources are required for all features types. Success in ongoing efforts in knowledge base construction projects, will make performance of our algorithm better.

Relation	Prep.	Attachment accuracy	Example(s)
acquired	from	**99.97**	BNY Mellon *acquired* Insight *from* Lloyds.
hasSpouse	in	**91.54**	David *married* Victoria *in* Ireland.
worksFor	as	**99.98**	Shubert *joined* CNN *as* reporter.
playsInstrument	with	**98.40**	Kushner *played* guitar *with* rock band Weezer.

Table 6: Binary relations extended to ternary relations by mapping to verb-preposition pairs in PP 5-tuples. PPAD predicted verb attachments with accuracy >90% in all relations.

5.5 Application to Ternary Relations

Through the application of ternary relation extraction, we further tested PPAD's PP disambiguation accuracy and illustrated its usefulness for knowledge base population. Recall that a PP 5-tuple of the form $\{n0, v, n1, p, n2\}$, whose enclosed PP attaches to the verb v, denotes a ternary relation with arguments *n0, n1, & n2*. Therefore, we can extract a ternary relation from every 5-tuple for which our method predicts a verb attachment. If we have a mapping between verbs and binary relations from a knowledge base (KB), we can extend KB relations to ternary relations by augmenting the KB relations with a third argument $n2$.

We considered four KB binary relations and their instances such as $worksFor(TimCook, Apple)$, from the NELL KB. We then took the collection of 4 million 5-tuples that we extracted from Wikipedia. We mapped verbs in 5-tuples to KB relations, based on significant overlaps in the instances of the KB relations, noun pairs such as $(TimCook, Apple)$ with the $n0, n1$ pairs in the Wikipedia PP 5-tuple collection. We found that, for example, instances of the noun-noun KB relation "worksFor" match $n0, n1$ pairs in tuples where $v = joined$ and $p = as$, with $n2$ referring to the job title. Other binary relations extended are: "hasSpouse" extended by "in" with wedding location, "acquired" extended by "from" with the seller of the company being acquired. Examples are shown in Table 6. In all these mappings, the proportion of verb attachments in the corresponding PP quads is significantly high ($> 90\%$). PPAD is overwhelming making the right attachment decisions in this setting.

Efforts in temporal and spatial relation extraction have shown that higher N-ary relation extraction is challenging. Since prepositions specify details that transform binary relations to higher N-ary relations, our method can be used to read information that can augment binary relations already in KBs. As future work, we would like to incorporate our method into a pipeline for reading beyond binary relations. One possible direction is to read details about the *where,why, who* of events and relations, effectively moving from extracting only binary relations to reading at a more general level.

6 Conclusion

We have presented a knowledge-intensive approach to prepositional phrase (PP) attachment disambiguation, which is a type of syntactic ambiguity. Our method incorporates knowledge about verbs, nouns, discourse, and noun-noun binary relations. We trained a model using labeled data and unlabeled data, making use of expectation maximization for parameter estimation. Our method can be seen as an example of tapping into a positive feedback loop for machine reading, which has only become possible in recent years due to the progress made by information extraction and knowledge base construction techniques. That is, using background knowledge from existing resources to read better in order to further populate knowledge bases with otherwise difficult to extract knowledge. As future work, we would like to use our method to extract more than just binary relations.

Acknowledgments

We thank Shashank Srivastava and members of the NELL team at CMU for helpful comments. This research was supported by DARPA under contract number FA8750-13-2-0005.

References

Eneko Agirre, Timothy Baldwin, and David Martinez. 2008. Improving parsing and PP attachment performance with sense information. In *Proceedings of ACL-08: HLT*, pages 317–325.

Gerry Altmann and Mark Steedman. 1988. Interaction with context during human sentence processing. *Cognition*, 30:191–238.

Enrique Henestroza Anguiano and Marie Candito. 2011. Parse correction with specialized models for difficult attachment types. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 1222–1233.

Michaela Atterer and Hinrich Schütze. 2007. Prepositional phrase attachment without oracles. *Computational Linguistics*, 33(4):469–476.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, pages 722–735.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250.

Eric Brill and Philip Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *15th International Conference on Computational Linguistics, COLING*, pages 1198–1204.

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka, Jr., and Tom M. Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 101–110.

Michael Collins and James Brooks. 1995. Prepositional phrase attachment through a backed-off model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 27–38.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Recources and Evaluation (LREC*, pages 449–454.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: Clause-based open information extraction. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 355–366.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011a. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011b. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press.

Lyn Frazier. 1978. *On comprehending sentences: Syntactic parsing strategies*. Ph.D. thesis, University of Connecticut.

Sanda M. Harabagiu and Marius Pasca. 1999. Integrating symbolic and statistical methods for prepositional phrase attachment. In *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society ConferenceFLAIRS*, pages 303–307.

Donald Hindle and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

Dirk Hovy, Ashish Vaswani, Stephen Tratz, David Chiang, and Eduard Hovy. 2011. Models and training for unsupervised preposition sense disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, pages 323–328.

John Kimball. 1988. Seven principles of surface structure parsing in natural language. *Cognition*, 2:15–47.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42(1):21–40.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics,ACL*, pages 423–430.

Ni Lao, Tom Mitchell, and William W Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 236–244.

Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Tanti Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2302–2310.

Ndapandula Nakashole and Tom M. Mitchell. 2014. Language-aware truth assessment of fact candidates. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1009–1019.

Ndapandula Nakashole and Gerhard Weikum. 2012. Real-time population of knowledge bases: opportunities and challenges. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 41–45. Association for Computational Linguistics.

Ndapandula Nakashole, Martin Theobald, and Gerhard Weikum. 2011. Scalable knowledge harvesting with high precision and high recall. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 227–236.

Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. 2013. Fine-grained semantic typing of emerging entities. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL*, pages 1488–1497.

Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

Patrick Pantel and Dekang Lin. 2000. An unsupervised approach to prepositional phrase attachment using contextually similar words. In *38th Annual Meeting of the Association for Computational Linguistics, ACL*.

Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 250–255.

Adwait Ratnaparkhi. 1998. Statistical models for unsupervised prepositional phrase attachement. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL*, pages 1079–1085.

Vivek Srikumar and Dan Roth. 2013. Modeling semantic relations expressed by prepositions. *TACL*, 1:231–242.

Jiri Stetina and Makoto Nagao. 1997. Prepositional phrase attachment through a backed-off model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 66–80.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Kristina Toutanova, Christopher D. Manning, and Andrew Y. Ng. 2004. Learning random walk models for inducing word dependency distributions. In *Machine Learning, Proceedings of the Twenty-first International Conference, ICML*.

Olga van Herwijnen, Antal van den Bosch, Jacques M. B. Terken, and Erwin Marsi. 2003. Learning PP attachment for filtering prosodic phrasing. In *10th Conference of the European Chapter of the Association for Computational Linguistics,EACL*, pages 139–146.

Greg Whittemore, Kathleen Ferrara, and Hans Brunner. 1990. Empirical study of predictive powers od simple attachment schemes for post-modifier prepositional phrases. In *28th Annual Meeting of the Association for Computational Linguistics,ACL*, pages 23–30.

Derry Wijaya, Ndapandula Nakashole, and Tom Mitchell. 2014. Ctps: Contextual temporal profiles for time scoping facts via entity state change detection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Shaojun Zhao and Dekang Lin. 2004. A nearest-neighbor method for resolving pp-attachment ambiguity. In *Natural Language Processing - First International Joint Conference, IJCNLP*, pages 545–554.