Machine Reading with High Volume Memory

Nakashole NDAPA@CS.CMU.EDU
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15213

Tom M Mitchell TOM.MITCHELL@CS.CMU.EDU
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15213

Abstract

In artificial intelligence, the goal of machine reading is to develop systems that automatically understand natural language text. A key challenge in machine reading arises from a significant amount information not being explicitly stated but instead implied by the text in combination with background knowledge. This means that machine reading methods that rely on context alone are inherently limited in their understanding capabilities. In this paper, we advocate and study machine reading methods that incorporate comprehensive, high volume world knowledge in their inference mechanisms. To this end, we have developed methods for sentence level machine reading that make use of background knowledge. The first method addresses prepositional phrase attachment ambiguity. Our method uses background knowledge within a semi-supervised machine learning algorithm that learns from both labeled and unlabeled data. The approach produced state-of-the-art results on two datasets and performed significantly better than a widely used dependency parser. The second method aims to extract relationships from compound nouns. We have developed a knowledge-aware method for compound noun analysis, our experiments show that it accurately extracts beliefs from compound nouns.

1. Introduction

Artificial intelligence researchers have long sought to build systems capable of automatically reading understanding natural language text – machine reading systems. A computer program is said to understand language if it responds appropriately to instructions in natural language. For example, if the task is to read a piece of text and answer questions about it, then a program understands if it outputs the correct answers. If the task is to translate from one language to another, the program understands if it correctly translates from the source language to the target language. And if the task is to extract information about which drugs treats which physiological conditions, the program understands if it finds the correct pairs of drugs and physiological conditions.

Machine reading methods can be characterized based on their awareness of world knowledge. On one extreme end, there are methods that are oblivious to background knowledge, *reading from scratch methods*. On the opposite end, there are knowledge-intensive methods that incorporate comprehensive, high volume world knowledge in their inference mechanisms, *reading with high volume memory*. A key challenge in language understanding is that some information is not explicitly stated in text but it is implied by the text in combination with background knowledge. For example, if the text states that Alice left a restaurant after a good meal, one can infer, with some probability, that she paid the bill and left a tip. In reference resolution, consider the sentence "The bee landed on the flower because it wanted pollen." If we know that bees feed on pollen, we can correctly determine that "it" refers to the bee and not the flower. In negation detection, consider the sentence: "Things would be different if Microsoft was headquartered in Texas." From this sentence alone, a machine reading program might incorrectly extract a belief that Microsoft is headquartered in Texas. But from the prior knowledge that Microsoft was never headquartered in Texas, we might be able to better detect the negation, in addition to the syntactic cues such as "if". Thus, inference over prior knowledge is crucial to text understanding. When humans read, this kind of inference is natural. Studies of brain scans of people's brains while reading fiction have found that readers mentally simulate each new situation encountered in a story(Wehbe, Vaswani, Knight, & Mitchell, 2014). Details about actions and sensation are captured from the text and integrated with personal knowledge from past experiences.

In this paper, we study machine reading methods that leverage a high volume memory of beliefs about the world. In particular, we have developed two methods for sentence level machine reading that make use of background knowledge:

1. The first method addresses a difficult case of syntactic ambiguity caused by prepositions. Prepositions such as "in", "at", and "for" express important details about the where, when, and why of relations and events. However, prepositions are major source of syntactic ambiguity and still pose problems in language analysis. In particular, they cause the problem of prepositional phrase attachment ambiguity, which arises in cases such as "she caught the butterfly with the spots" vs. "she caught the butterfly with the net". In the first case, the preposition "with" modifies the verb "caught", while in the second case, "with" modifies the noun "butterfly". Disambiguating these two attachments requires knowing that butterflies can have spots, and that a net is an instrument that can be used for catching. Our approach uses this type of knowledge within a semi-supervised machine learning algorithm that learns from both labeled and unlabeled data. The approach produced state-of-the-art results on two datasets and performed significantly better than a syntactic parser that most people use in their natural language processing pipelines.

2. The second method that exploits background knowledge for language understanding extracts relationships from compound nouns such as "pro-choice Democratic gubernatorial candidate James Florio", or "White House spokesman Marlin Fitzwater". Compound nouns contain a number of challenging compositional phenomena, including implicit relations. Compound nouns mostly consist of adjectives and nouns, they do not contain verbs. That means there are often no lexical commonalities even across compound nouns that express the same relations, making it difficult to extract beliefs from them. On the other hand, beliefs such as a person's job title, nationality, or stance on a political issue are often expressed using compound nouns. We have developed a knowledge-aware method for compound noun analysis which accurately extracts relationships from compound nouns.

1.1 High Volume Memory

By high volume memory we mean storage capable of storing and retrieving comprehensive world knowledge akin to the breadth of world knowledge adult human brains have a grasp of. Such a high volume memory, therefore, contains beliefs about the world, the objects in it, their properties, and approximate confidence scores for beliefs held.

Knowledge Bases. Towards realizing high volume memories of world knowledge, knowledge base construction projects have accumulated large amounts of beliefs about real world entities (Mitchell, 2015; Suchanek, Kasneci, & Weikum, 2007; Bollacker, Evans, Paritosh, Sturge, & Taylor, 2008). Large-scale knowledge bases are populated by applying machine reading methods to web corpora. For example, the Never-Ending Language Learner (NELL) system has been learning to read the web 24 hours/day for over several years. However, current machine reading methods have been successful at populating knowledge bases by means of pattern detection— a shallow way of machine reading which leverages the redundancy of large corpora to capture language patterns. However, machine readers still lack the ability to fully understand language. In the pursuit of the much harder goal of language comprehension, knowledge bases present an opportunity for a virtuous circle: the accumulated knowledge can be used to improve machine readers; in turn, advanced reading methods can be used to populate knowledge bases with beliefs expressed using complex and potentially ambiguous language. There has been little work on making use of knowledge bases in machine reading. (Krishnamurthy & Mitchell, 2014) introduced a method for training a joint syntactic and semantic parser. Their parser makes use of a knowledge base to produce logical forms containing knowledge base predicates. However, their use of the knowledge base is limited to unary predicates to determine semantic types of concepts. In contrast, in this paper we make extensive use of a knowledge base augmented by linguistic resources and corpus statistics as the content of a high volume memory that our knowledge-aware methods have access to at inference time.

Neural Networks with Long-Term Memory. Recent years have seen wide use of neural network models for language understanding. For example neural networks have been applied to the task of answering queries about short stories, and to the task of language modeling where the task is to predict the next word(s) in a text sequence given the previous words (Mikolov, Karafiát, Burget, Cernocký, & Khudanpur, 2010; Sundermeyer, Schlüter, & Ney, 2012; Weston, Chopra, & Bordes, 2015; Sukhbaatar, Weston, Fergus, et al., 2015). These tasks are treated as instances of sequence processing, where the sequence consists of sentences or in the case of language modeling the sequence consists of words. A number of studies have explored neural networks that model long-term structure in sequences using recurrent neural networks (RNNs) including Long short-term memory (LSTMs) (Hochreiter & Schmidhuber, 1997; Atkeson & Schaal, 1995; Graves, 2013). However, the memory in these models is the state of the network which is encoded using latent states and weights. This memory is therefore typically too small and not structured enough to remember facts from the past since background knowledge is compressed into dense vectors. Additionally, in neural approaches, to retrieve relevant memories, smooth lookups are performed, whereby each memory is scored for its relevance, this may not scale well to the case where a larger memory is required. A notable exception in this line of work is the work of (Weston et al., 2015) which introduced memory networks combining RNN inference with a long-term memory. They applied their models to the task of answering queries about short stories. However background knowledge in this work is raw text from the story as opposed to high volume world knowledge. While they also performed an experiment on a question answering setting with background knowledge consisting of 14M statements stored as (subject, relation, object) triples, this setting is not a machine reading task but an information retrieval task since there was no reading required to answer the questions, but only look up to find the triples most relevant to the question.

1.2 Contributions

Our main contributions are as follows:

1) Machine Reading with High Volume Memory: We describe the problem of machine reading with high volume memory. While the problem of machine reading has attracted a lot of attention in recent years, there's been very little work on machine reading with high volume memory. This setting is unique and raises new questions which we study within the context of two problems: prepositional phrase attachment, and compound noun relation extraction.

2) Prepositional Phrase Attachment: We present a knowledge-aware method for prepositional phrase attachment. Previous methods largely rely on corpus statistics. Our approach draws upon diverse sources of background knowledge, leading to performance improvements. 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. In addition to the standard Wall Street Journal corpus (WSJ) (Ratnaparkhi, Reynar, & Roukos, 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]. Although this work was first published in

3) Compound Noun Analysis: We introduce a knowledge-aware method for extracting relations from compound nouns. We collected over 2 million compound nouns from which we learned fine-grained semantic type sequences that express relations from the NELL knowledge base. Our experiments show that the accuracy of relations extracted in this manner is quite high.

1.3 Organization

The rest of the paper is organized as follows. We begin by presenting our knowledge-aware methods for machine reading. Section 2 presents our knowledge-aware approach to prepositional phrase attachment ambiguity. Section 3 introduces our knowledge-aware approach to relation extraction from compound nouns. Lastly, Section 4 provides concluding remarks and future directions.

2. Prepositional Phrase Attachment

Prepositional phrases (PPs) express crucial information that information extraction 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.* 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: $\langle$Alice$\rangle$ caught $\langle$butterfly with spots$\rangle$. 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: $\langle$Alice$\rangle$ caught $\langle$butterfly$\rangle$ with $\langle$net$\rangle$.

[1] http://rtw.ml.cmu.edu/resources/ppa

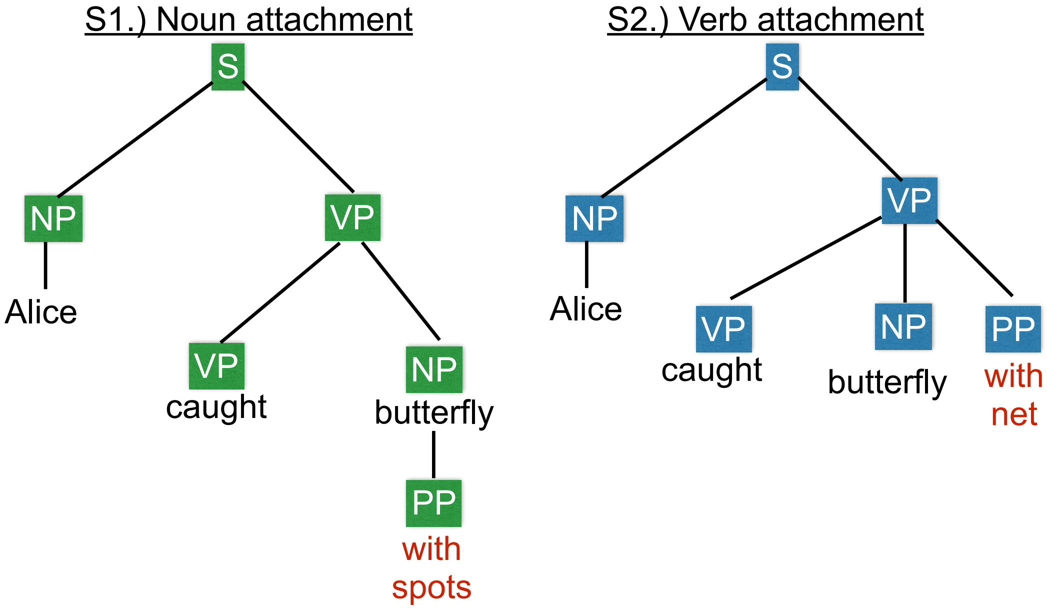


Figure 1: Parse trees where the prepositional phrase (PP) attaches to the noun, and to the verb.

The PP attachment problem is often defined as follows: given a PP occurring within a sentence where there are multiple possible attachment sites for the PP, choose the most plausible attachment site. In the literature, prior work going as far back as (Brill & Resnik, 1994; Ratnaparkhi et al., 1994; Collins & 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, Laird, & Rubin, 1977).

2.1 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 & 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 per-

[2] http://nlp.stanford.edu:8080/parser/

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.

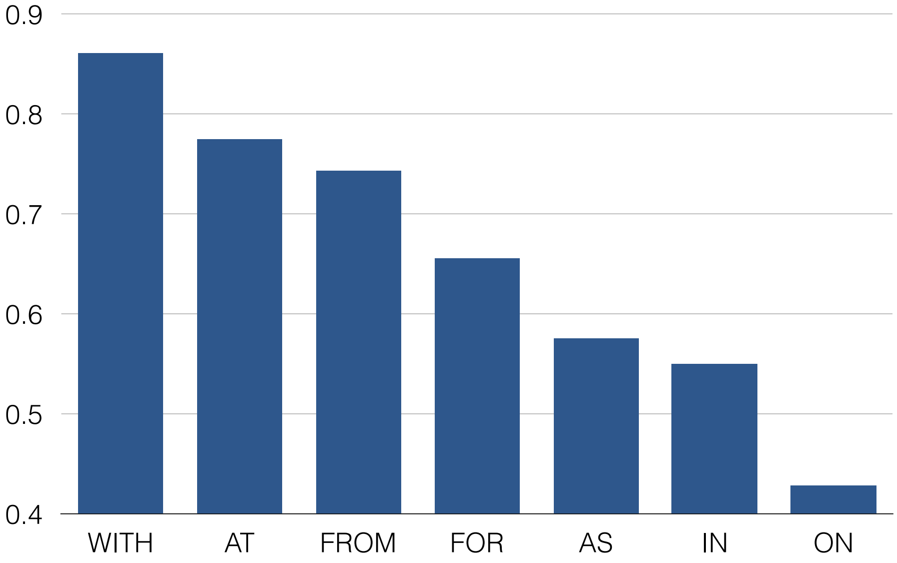


Figure 2: Dependency parser PP attachment accuracy for various frequent prepositions.

forms 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.

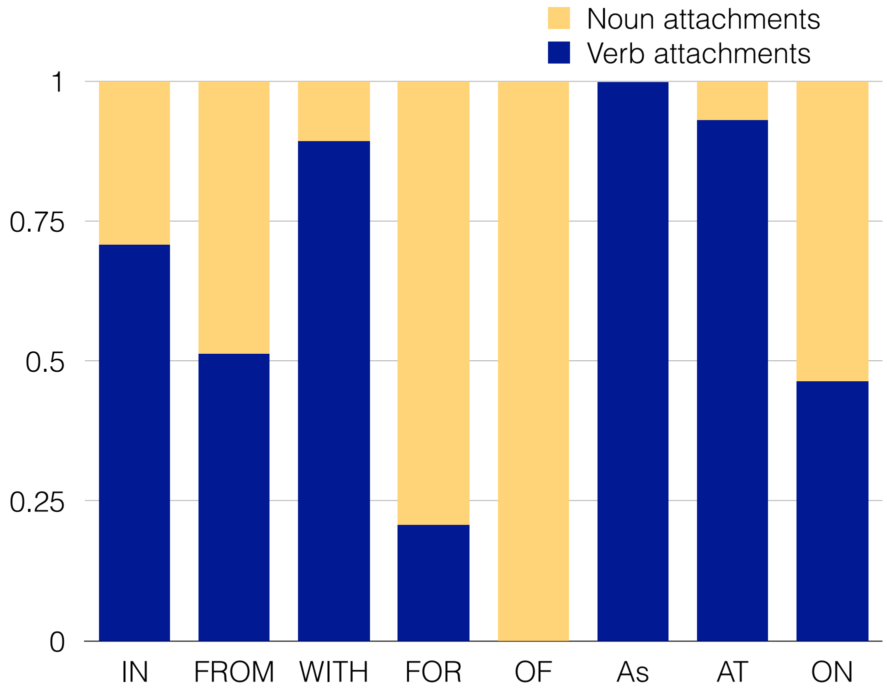


Figure 3: Noun vs. verb attachment proportions for frequent prepositions in the labeled NYTC dataset.

2.2 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 & Brooks, 1995; Brill & Resnik, 1994) or from automatically acquired training data that may be noisy (Ratnaparkhi, 1998; Pantel & 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 & 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 & Nagao, 1997; Toutanova, Manning, & Ng, 2004) to obtain less sparse corpus statistics. Corpus-derived clusters of similar nouns and verbs have also been used (Pantel & Lin, 2000).

Hindle and Rooth proposed a lexical association approach based on how words are associated with each other (Hindle & 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, Ferrara, & Brunner, 1990).

Rule-based Methods. (Brill & 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 & Schütze, 2007; Agirre, Baldwin, & Martinez, 2008; Anguiano & 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, Vaswani, Tratz, Chiang, & Hovy, 2011; Srikumar & 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.

2.3 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:

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.

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: $q1 = \{caught, butterfly, with, net\} : V$, $q2 = \{caught, butterfly, with, spots\} : N$.

$$\begin{array}{l} \textit{(Verb attach)} \longrightarrow v\ \langle \textit{has-slot-filler} \rangle\ n2 \\ \hline \textit{(Noun attach a.)} \longrightarrow n1\ \langle \textit{described-by} \rangle\ n2 \\ \textit{(Noun attach b.)} \longrightarrow n2\ \langle \textit{described-by} \rangle\ n1 \end{array}$$

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

2.3.1 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.

2.3.2 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).

2.3.3 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, Korhonen, Ryant, & Palmer, 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.

2.3.4 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 $F2$.

2.3.5 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 & Brooks, 1995).

2.4 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 earlier..

2.4.1 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.

2.4.2 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:

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

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:

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

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

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

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.

2.5 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 $n1$.

2.5.1 Experimental Setup

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

Algorithm 1 The EM algorithm for PP attachment

```
Input: $\mathcal{X}, \mathcal{D} = D^l \cup D^u$
Output: $\theta^T$
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, $\theta^t$
  $\theta^{\mathbf{t}} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta^{\mathbf{t-1}})$
      $\mathcal{Q}(\theta, \theta^{t-1}) = \sum_{i=1}^{N} \sum_{y \in \{0,1\}} p(y|x_i, \theta^{t-1}) \times$
                                                  $(y\theta^T x_i - \ln(1 + exp(\theta^T x_i)))$
  if convergence($\mathcal{L}(\theta), \mathcal{L}(\theta^{t-1})$) then
    break
  end if
end for
return $\theta^T$
```

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.

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 & 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.

	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.

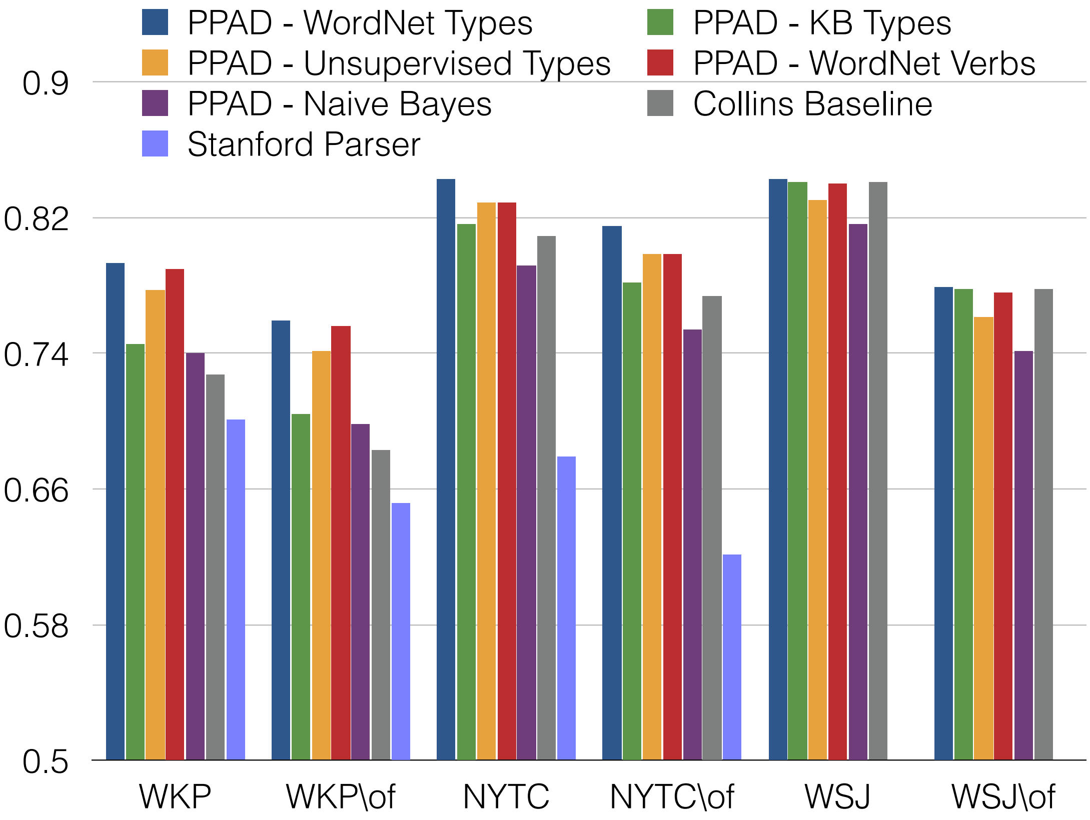


Figure 4: PPAD variations vs. baselines.

2.5.2 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.

2.5.3 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, 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 2.5.1, PPAD corresponds to the best model.

2.5.4 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

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

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.

2.5.5 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 $n2$.

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.

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.

2.5.6 LABELED TERNARY ARGUMENTS

In the above experiment, we studied the case of extending existing KB relations to ternary relations with a third argument. However, we did not have any semantic information about the role of the third arguments. In this section, we study a different case, the case when we want to label the role of the third argument. For example, for the acquisition instance of "BNY Mellon acquired Insight from Lloyds", we want to predict that the label of "Lloyds" is the "Source", indicating the source company of acquisition. As another example, consider the buy instance 'Bailey bought earrings for Josie", we want to predict that the label of "Josie" is "Beneficiary", indicating the beneficiary of the earrings bought.

To obtain labels for third arguments, we make use of VerbNet (Kipper et al., 2008). VerbNet provides, for each verb, frames of the different use cases of the verb. Here we consider only verb uses that make use of prepositions. In VerbNet, these frames are described using a label of "primary=NP V NP PP.label" where the "label" is the role of the third arguments following the prepositional phrase. One example is "primary=NP V NP PP.instrument", each such frame is accompanied by an example sentence. In this case the example is: "Paula hit the ball with a stick", where the "stick" takes the role of the instrument. Notice that a given verb and preposition combination does not necessary invoke a given label. For example in "Paula hit the ball with joy", "joy" does not play the role of the instrument. Therefore, we learn introduce further constraints. We learn these constraints from the collection of 4 million 5-tuples that we extracted from Wikipedia as explained in Section 2.5.1. In particular, we replace mentions of entities with their NELL and WordNet semantic types. Using this approach, we generate templates of the form:

<np_v_np_pp.LABEL ><verb><typeofArg1><preposition ><typeofArg2>

We worked with five labels from VerbNet: np_v_np_pp.beneficiary, np_v_np_pp.instrument, np_v_np_pp.asset, np_v_np_pp.source, and np_v_np_pp.topic. Examples of learned templates for each of the five labels are as shown below in Table 7.

Table 8 shows sample instances of the different learned templates for labeled ternary arguments. We randomly sampled 100 such instances evaluated them for accuracy, we found a sampling accuracy of 88%.

<np_v_np_pp.beneficiary ><buy><jewelry><for ><person>
<np_v_np_pp.instrument ><shoot><person><with ><weapon>
<np_v_np_pp.asset ><sell><company><for ><amount>
<np_v_np_pp.source ><buy><organization><from ><organization>
<np_v_np_pp.topic ><ask><person><for ><advice>
<np_v_np_pp.topic ><ask><person><for ><divorce>

Table 7: Examples of learned templates for labeled ternary relations

2.6 Prepositional Phrase Attachment Ambiguity Summary

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.

Ternary argument label	Instance
np_v_np_pp.beneficiary	danai udomchoke won gold medal for thailand
np_v_np_pp.beneficiary	alton cooked breakfast for crew
np_v_np_pp.beneficiary	boys cooked cakes for girls
np_v_np_pp.beneficiary	bailey buys earrings for josie
np_v_np_pp.beneficiary	jim buys bracelet for kathy
np_v_np_pp.beneficiary	leonard buys engagement ring for michelle
np_v_np_pp.beneficiary	headmaster bought goggles for children
np_v_np_pp.instrument	lord edward thynne shot golden eagle with rifle
np_v_np_pp.instrument	mohawks opened fire with gunshots
np_v_np_pp.instrument	unidentified militants opened fire with grenade launcher
np_v_np_pp.instrument	jarvis opened fire with 5-inch guns
np_v_np_pp.instrument	prince stabs vizier with dagger
np_v_np_pp.instrument	isaac van scoy killed british soldier with pitchfork
np_v_np_pp.instrument	ambush positions opened fire with mortars
np_v_np_pp.instrument	tamalika karmakar killed rebecca with knife
np_v_np_pp.source	telugu film homam drew inspiration from martin scorsese
np_v_np_pp.source	john coltrane received call from davis
np_v_np_pp.source	kenneth o'keefe received letter from state department
np_v_np_pp.source	tony receives letter from mandy
np_v_np_pp.source	peter receives call from claire
np_v_np_pp.source	huppertz drew inspiration from richard wagner
np_v_np_pp.source	fiz receives call from alan hoyle
np_v_np_pp.source	elbaz drew inspiration from bruce willis
np_v_np_pp.source	smolensky bought company from wheeler
n np_v_np_pp.topic	wittenberg asked jan kazimierz for permission
np_v_np_pp.topic	brando asked john gielgud for advice
np_v_np_pp.topic	lutician delegates asked conrad for help
np_v_np_pp.topic	logan asked scott for help
np_v_np_pp.topic	philadelphia quakers asked nhl for permission
np_v_np_pp.topic	steven asks frank for advice
np_v_np_pp.topic	rowe asked jackson for divorce

Table 8: Sample instances of templates learned for labeled ternary arguments. For each instance, the label applies to the last argument.

3. Compound Nouns Analysis

The second example of a language construct that causes problems for machine reading in the absence of background knowledge arises from the use of compound nouns. Noun phrases contain a number of challenging compositional phenomena, including implicit relations. Compound nouns such as “pro-choice Democratic gubernatorial candidate James Florio”, or “White House spokesman Marlin Fitzwater” primarily consist of nouns and adjectives. They do not contain verbs. This means that it is difficult for a pattern detection algorithm to detect any useful lexical regularities across compound nouns that express the same relations. On the other hand, beliefs such as a persons job title, nationality, or stance on a political issue are often expressed using compound nouns. We propose a knowledge-aware algorithm for extracting semantic relations from compound noun analysis that learns, through distant supervision, to map fine-grained type sequences of compound nouns to the relations they express. Consider the following compound nouns.

1.a) Giants cornerback Aaron Ross
1.b) Patriots quarterback Matt Cassel
1.c) Colts receiver Bryan Fletcher

2.a) Japanese astronaut Soichi Noguchi
2.b) Irish golfer Padraig Harrington
2.c) French philosopher Jean-Paul Sartre

2.a) Seabiscuit author Laura Hillenbrand
2.b) Harry porter author J.K Rowling
2.c) Walking the Bible author Bruce Feile

The concepts in the compound noun sequences *(1a. – c.)*, *(2a. – c.)*, *(3a. – c.)* are of the semantic type sequences:

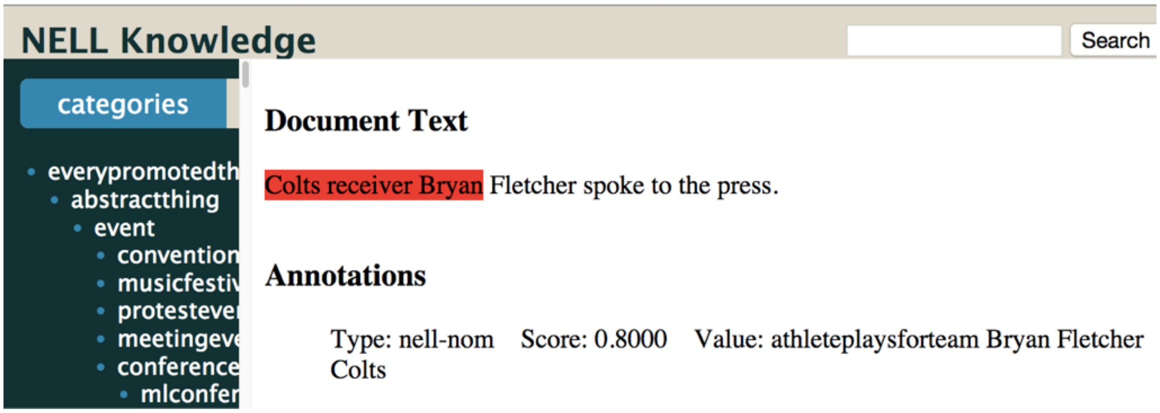
<sportsteam><sportsteamposition><athlete>
<country><profession><person>
<book>“author” <person>

Therefore, our task is to learn semantic type sequences and their mappings to knowledge base relations. In our case we use relations from the NELL knowledge base. Since NELL has binary relations that take only two arguments, and compound nouns contain more than two noun phrases, we additionally keep track of the position information for the two arguments of the relation. For example, from the type sequence: *<country ><profession><person>*, we generate mappings to two different relations.

Relation	**arg1_pos**	**arg2_pos**	**type sequence**
citizenofcountry	3	1	*<country><profession><person>*
personhasjobposition	3	2	*<country><profession><person>*

Table 9: Learned mappings from compound nouns semantic type sequences to binary relations

To learn mappings from compound nouns to binary relations as shown in Table 9, we use distant supervision, that is using the NELL knowledge base as the only form of supervision. In general,



Document Text

Colts receiver Bryan Fletcher spoke to the press.

Annotations

Type: nell-nom Score: 0.8000 Value: athleteplaysforteam Bryan Fletcher Colts

Figure 5: Extracting the athleteplaysforteam relation from a compound noun.

the intuition behind distant supervision is that a sentence that contains a pair of entities that participate in a known knowledge relation is likely to express that relation. In our case, the sentence is just a compound noun. Therefore, we first extract compound nouns from a large collection of documents. For every compound noun, we map its noun phrase to entities in the NELL. The entities are then replaced by their NELL types. This creates type sequences of the form: <*country* ><*profession*><*person*>. Each type sequence has a support set, which is the collection of compound nouns that satisfy the type sequence. For example, a support compound noun might be: *Japanese astronaut Soichi Noguchi* for the type sequence: <*country*><*profession*><*person*>. We retain type sequences whose support set sizes are above a threshold of 10 in our experiments. For type sequences with support set size above the threshold, we use their support sets to learn mappings from type sequences to relations using distant supervision. That is, from each supporting compound noun we collect pairs of entities, and do look ups in NELL to determine which relations hold between the pair of entities. We additionally keep track of the position of the entities within the compound noun. This gives us mappings from types sequences to relations such as: <*citizenofcountry*><*3*><*1*><*country*><*profession*>. We only retain mappings that have a support set size (relation instances in NELL) above a threshold of 10 in our experiments.

3.1 Experimental Evaluation

We extracted compound nouns from three different corpora: the New York Times archive which includes about 1.8 Million articles from the years 1987 to 2007, the English edition of Wikipedia with about about 3.8 Million articles, and the KBP dataset (Surdeanu, 2013) which contains over 2 million documents with Gigaword newswire and Wb documents. We extracted a total of $2,270,487$ compound nouns. From these compound nouns, we extract 10 relations that are expressed by compound nouns and are in the NELL knowledge base. From this dataset we learned 291 mappings from types sequences to relations. Using these mappings we then predicted new relation instances. We report recall and accuracy in Table 10. We can see that this approach has high accuracy across all the relations. Recall is also high for certain relations but low for others. This might be because certain relations although they are expressed using compound nouns, they are more commonly expressed in other forms.

We have incorporated this system into the NELL reading software. A screenshot of extracting relations from compound nouns is shown in Figure 5.

Relation	Recall	Precision
citizenofcountry	15,805	0.982 ± 0.018
citylocatedincountry	1,521	0.75 ± 0.083
athleteplaysforteam	471	0.982 ± 0.018
persongraduatedfromuniversity	49	0.964 ± 0.036
personhasjobposition	511,937	0.837 ± 0.07
musicianplaysinstrument	3,890	0.885 ± 0.06
worksfor	75,757	0.847 ± 0.068
athletewinsawardtrophytournament	92	0.928 ± 0.049
coachesteam	175	0.982 ± 0.018
companysubsidiary	37	0.953 ± 0.047

Table 10: Recall and precision of relation extraction for 10 relations. Precision is sampled from a total of max(100, recall).

3.2 Related Work

Much of the work on information extraction has been on extracting relations expressed by verb phrases that occur between pairs of noun phrases. Extracting knowledge base relations from noun phrases alone has been much less explored. In (Yahya, Whang, Gupta, & Halevy, 2014), a method is developed that learns noun phrase structure for open information extraction. This is different from our work in that we are extract knowledge base relations as opposed to open information extraction. Therefore, the authors do not ground their extracted attributes to an external knowledge base.

The work of (Choi, Kwiatkowski, & Zettlemoyer, 2015) developed a semantic parser for extracting relations from noun phrases. Given an input noun phrases, it is first transformed to a logical form, where the logical form is an intermediate unambiguous representation of the noun phrase. The logical form is chosen such that it closely matches the linguistic structure of the input text noun phrase. The logical form is then transformed into one that, where possible, uses the Freebase ontology predicates (Bollacker et al., 2008). These predicates can then be read off as relations expressed about the entities described by the noun phrase. The authors test their work on Wikipedia category names. Since each Wikipedia category describes a set of entities, by extracting relations from each category name, one learns relations about all the members of the category. Consider the Wikipedia category *Symphonic Poems by Jean Sibelius*. An example of the knowledge base transformed logical form for this category name would be:

$\lambda x.composition.form(x; Symphonic\,poems) \wedge composer(Jean\ Sibelius; x)$ where one can now extract attributes for the entities, such as *The Bard, Finlandia, Pohjolas Daughter, En Saga, Spring Song, Tapiola, …*, that fall under this category in particular that for all x in this category $composer(Jean\ Sibelius; x)$ and $composition.form(x; Symphonic\,poems)$ where $composer$ and $composition.form$ are Freebase attributes. In generating the logical forms, several features are used that capture some background knowledge, in particular, a number of features that enable soft type checking on the produced logical form, and features that test agreement of these types on different parts of the produced logical form.

In a related but different line of work, the NomBank project (Meyers, Reeves, Macleod, Szekely, Zielinska, Young, & Grishman, 2004; Gerber & Chai, 2010) annotated the argument structures for common nouns. For example, from the expression *Greenspans replacement Ben Bernanke*, the arguments for the nominal "replacement", are: "Ben Bernanke" is ARG0 and "Greenspan" is ARG1. The resulting annotations has been used as training data for work on semantic role labeling on nominals (Jiang & Ng, 2006; Liu & Ng, 2007). Again, this work is different from our work in that no knowledge base relations are extracted.

There has also work on the broader topic of semantic structure of noun phrases. In (Sawai, Shindo, & Matsumoto, 2015), a method is proposed that parsers noun phrases into the Abstract Meaning Representation in order to detect the argument structures, and noun-noun relations in compound nouns.

3.3 Compound Noun Analysis Summary

We have presented a knowledge-aware method for relation extraction from compound nouns. Our method uses semantic types of concepts in compound noun sequences to predict relations expressed by novel compound noun sequences containing concepts that we have not seen before. This method can be seen as another example of tapping into a positive feedback loop for machine reading made possible by projects that construct large-scale knowledge bases. Compound nouns are non-trivial to interpret in many different ways besides the noun-noun relations problem we addressed here. For example, one problem that could benefit from background knowledge is that of analyzing the internal structure of noun phrases through bracketing (Vadas & Curran, 2007, 2008). For example, in the noun phrase *(lung cancer) deaths*, the task would be to determine that *lung cancer* modifies the head *deaths*. Additionally, as future work we can increase our predicate vocabulary to learn more common sense type of relations from compound nouns, for example, in the noun phrase *cooking pot*, we can extract the relationpurpose, to mean the pot is used for cooking.

4. Discussion and Conclusion

In this paper, we presented results on two cases studies of using background knowledge: firs,t for prepositional phrase attachment ambiguity, we made use of several types of relevant background knowledge including binary relations, semantic types, and verb role filler information from VerbNet; second, we made use semantic types of concepts to extract relations from compound nouns. While these results show the potential of using high volume memory in machine reading methods, our experience also suggests there are crucial building blocks that need to be in place for this approach to be broadly applicable to machine reading systems beyond single language constructs such as compound nouns and prepositional phrases:

Broad Coverage Background Knowledge. The representation of knowledge found in knowledge bases is suitable for reasoning in machine reading learning mechanisms because it is tied to formal semantics and is typically free of inconsistencies. However, the mechanisms for building knowledge bases still have coverage limitations. For example, the NELL knowledge graph contains 1.34 facts per entity (Hegde & Talukdar, 2015). This knowledge sparsity curtails the performance gains we can obtain from knowledge-aware features. In the future, it will be beneficial to make use of knowledge-on-demand methods for acquiring knowledge, whereby we can pursue targeted knowledge harvesting at both training and test time as needed by our methods.

Learning methods. In this work we have used learning methods that treat the problem as linear function approximation problem with knowledge-aware features represented by sparse vectors. This representation of background knowledge is limited due to the constraint of the linear function. One direction for future work is to approximate more complex functions that are non-linear, and also to represent our knowledge-aware features as dense vectors.

Context Modeling. Understanding a piece of writing requires not only drawing upon background knowledge, but also upon discourse context. Instead of reading each sentence of a document as a self-contained unit, a machine reading program needs to keep track of what has been stated in preceding sentences. This is useful for dealing with basic language concepts such as entity co-reference, but also for keeping track of concepts already mentioned. Consider the sentence: “John saw the girl with the binoculars”. In the absence of context, the likely interpretation is that John used the binoculars to see the girl. However, if context suggests that there is a girl in possession of binoculars, the interpretation of the sentence changes. In the current work, we completely ignore context. Therefore, one direction for future work is explore how background knowledge interacts with context.

Acknowledgments

This research was supported by DARPA under contract number FA8750-13-2-0005. Any opinions, findings, conclusions and recommendations expressed in this paper are the authors’ and do not necessarily reflect those of the sponsor.

References

Agirre, E., Baldwin, T., & Martinez, D. (2008). Improving parsing and PP attachment performance with sense information. In *Proceedings of ACL-08: HLT*, pp. 317–325.

Altmann, G., & Steedman, M. (1988). Interaction with context during human sentence processing. *Cognition*, *30*, 191–238.

Anguiano, E. H., & Candito, M. (2011). Parse correction with specialized models for difficult attachment types. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pp. 1222–1233.

Atkeson, C. G., & Schaal, S. (1995). Memory-based neural networks for robot learning. *Neurocomputing*, *9*(3), 243–269.

Atterer, M., & Schütze, H. (2007). Prepositional phrase attachment without oracles. *Computational Linguistics*, *33*(4), 469–476.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. G. (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.*, pp. 722–735.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction for the web. In *IJCAI*, Vol. 7, pp. 2670–2676.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (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, pp. 1247–1250.

Brill, E., & Resnik, P. (1994). A rule-based approach to prepositional phrase attachment disambiguation. In *15th International Conference on Computational Linguistics, COLING*, pp. 1198–1204.

Carlson, A., Betteridge, J., Wang, R. C., Hruschka, Jr., E. R., & Mitchell, T. M. (2010). Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pp. 101–110.

Choi, E., Kwiatkowski, T., & Zettlemoyer, L. S. (2015). Scalable semantic parsing with partial ontologies. In *Association for Computational Linguistics (ACL)*.

Collins, M., & Brooks, J. (1995). Prepositional phrase attachment through a backed-off model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 27–38.

de Marneffe, M.-C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Recources and Evaluation (LREC*, pp. 449–454.

Del Corro, L., & Gemulla, R. (2013). Clausie: Clause-based open information extraction. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pp. 355–366.

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

Fader, A., Soderland, S., & Etzioni, O. (2011a). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pp. 1535–1545.

Fader, A., Soderland, S., & Etzioni, O. (2011b). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1535–1545. Association for Computational Linguistics.

Fellbaum, C. (Ed.). (1998). *WordNet: an electronic lexical database*. MIT Press.

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

Gerber, M., & Chai, J. Y. (2010). Beyond nombank: A study of implicit arguments for nominal predicates. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pp. 1583–1592.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Harabagiu, S. M., & Pasca, M. (1999). Integrating symbolic and statistical methods for prepositional phrase attachment. In *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society ConferenceFLAIRS*, pp. 303–307.

Hegde, M., & Talukdar, P. P. (2015). An entity-centric approach for overcoming knowledge graph sparsity. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 530–535.

Hindle, D., & Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics*, *19*(1), 103–120.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hovy, D., Vaswani, A., Tratz, S., Chiang, D., & Hovy, E. (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*, pp. 323–328.

Jiang, Z. P., & Ng, H. T. (2006). Semantic role labeling of nombank: A maximum entropy approach. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 138–145. Association for Computational Linguistics.

Kimball, J. (1988). Seven principles of surface structure parsing in natural language. *Cognition*, *2*, 15–47.

Kipper, K., Korhonen, A., Ryant, N., & Palmer, M. (2008). A large-scale classification of english verbs. *Language Resources and Evaluation*, *42*(1), 21–40.

Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics,ACL*, pp. 423–430.

Krishnamurthy, J., & Mitchell, T. M. (2014). Joint syntactic and semantic parsing with combinatory categorial grammar. In *ACL*.

Lao, N., Mitchell, T., & Cohen, W. W. (2011). Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 529–539. Association for Computational Linguistics.

Liu, C., & Ng, H. T. (2007). Learning predictive structures for semantic role labeling of nombank. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., & Grishman, R. (2004). Annotating noun argument structure for nombank. In *LREC*. European Language Resources Association.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *11th Annual Conference of the International Speech Communication Association, (INTERSPEECH)*.

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pp. 1003–1011. Association for Computational Linguistics.

Mitchell, J., & Lapata, M. (2008). Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, ACL*, pp. 236–244.

Mitchell, T. M. (2015). Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 2302–2310.

Nakashole, N., & Mitchell, T. M. (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*, pp. 1009–1019.

Nakashole, N., Theobald, M., & Weikum, G. (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, pp. 227–236.

Nakashole, N., Tylenda, T., & Weikum, G. (2013). Fine-grained semantic typing of emerging entities. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL*, pp. 1488–1497.

Nakashole, N., & Weikum, G. (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*, pp. 41–45. Association for Computational Linguistics.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*(2/3), 103–134.

Pantel, P., & Lin, D. (2000). An unsupervised approach to prepositional phrase attachment using contextually similar words. In *38th Annual Meeting of the Association for Computational Linguistics, ACL*.

Ratnaparkhi, A. (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*, pp. 1079–1085.

Ratnaparkhi, A., Reynar, J., & Roukos, S. (1994). A maximum entropy model for prepositional phrase attachment. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pp. 250–255.

Sawai, Y., Shindo, H., & Matsumoto, Y. (2015). Semantic structure analysis of noun phrases using abstract meaning representation. In *ACL (2)*, pp. 851–856.

Srikumar, V., & Roth, D. (2013). Modeling semantic relations expressed by prepositions. *TACL*, *1*, 231–242.

Stetina, J., & Nagao, M. (1997). Prepositional phrase attachment through a backed-off model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 66–80.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pp. 697–706. ACM.

Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. In *Advances in Neural Information Processing Systems*, pp. 2431–2439.

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. In *INTERSPEECH*, pp. 194–197.

Surdeanu, M. (2013). Overview of the TAC2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *TAC*. NIST.

Toutanova, K., Manning, C. D., & Ng, A. Y. (2004). Learning random walk models for inducing word dependency distributions. In *Machine Learning, Proceedings of the Twenty-first International Conference, ICML*.

Vadas, D., & Curran, J. R. (2007). Adding noun phrase structure to the penn treebank. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

Vadas, D., & Curran, J. R. (2008). Parsing noun phrase structure with CCG. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pp. 335–343.

van Herwijnen, O., van den Bosch, A., Terken, J. M. B., & Marsi, E. (2003). Learning PP attachment for filtering prosodic phrasing. In *10th Conference of the European Chapter of the Association for Computational Linguistics,EACL*, pp. 139–146.

Wehbe, L., Vaswani, A., Knight, K., & Mitchell, T. M. (2014). Aligning context-based statistical models of language with brain activity during reading. In *EMNLP*, pp. 233–243. ACL.

Weston, J., Chopra, S., & Bordes, A. (2015). Memory networks. In *In International Conference on Learning Representations, ICLR*.

Whittemore, G., Ferrara, K., & Brunner, H. (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*, pp. 23–30.

Wijaya, D., Nakashole, N., & Mitchell, T. (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.

Yahya, M., Whang, S., Gupta, R., & Halevy, A. Y. (2014). Renoun: Fact extraction for nominal attributes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 325–335. ACL.

Zhao, S., & Lin, D. (2004). A nearest-neighbor method for resolving pp-attachment ambiguity. In *Natural Language Processing - First International Joint Conference, IJCNLP*, pp. 545–554.