# Quantifier Scope Disambiguation Using Extracted Pragmatic Knowledge: Preliminary Results

**Prakash Srinivasan**
Temple University
1805 N. Broad St.
Wachman Hall 324
Philadelphia, PA 19122
`prakash.srinivasan@temple.edu`

**Alexander Yates**
Temple University
1805 N. Broad St.
Wachman Hall 324
Philadelphia, PA 19122
`yates@temple.edu`

## Abstract

It is well known that pragmatic knowledge is useful and necessary in many difficult language processing tasks, but because this knowledge is difficult to acquire and process automatically, it is rarely used. We present an open information extraction technique for automatically extracting a particular kind of pragmatic knowledge from text, and we show how to integrate the knowledge into a Markov Logic Network model for quantifier scope disambiguation. Our model improves quantifier scope judgments in experiments.

## 1 Introduction

It has long been a goal of the natural language processing (NLP) community to be able to interpret language utterances into logical representations of their meaning. Quantifier scope ambiguity has been recognized as one particularly challenging aspect of this problem. For example, the following sentence has two possible readings, depending on the scope of its quantifiers:

```
Every boy wants a dog.
```

One reading of this sentence is that there exists a single dog in the world which all boys want. The second, and usually preferred, reading is that the sentence is describing a separate "wanting" relation for each boy, and that the dog in question is a function of the boy who wants it. In this reading, there may be as many different dogs as boys, although it leaves open the possibility that several of the boys want the same dog. In logic, these two readings can be represented as follows:

1. $\exists_{d \in Dogs} \ \forall_{b \in Boys}$ wants$(b, d)$
2. $\forall_{b \in Boys} \ \exists_{d \in Dogs}$ wants$(b, d)$

The readings differ only in the order of the quantifiers. The quantifier that comes first in each expression is said to have *wide scope*; the second quantifier has *narrow scope*.

Linguists and NLP researchers have come up with several theories and mechanisms for automatically determining the scope of quantified linguistic expressions. Despite a long history of proposed solutions, however, researchers have for the most part abandoned this task as hopeless because of "overwhelming evidence suggesting that quantifier scope is a phenomenon that must be treated at the pragmatic level" (Saba and Corriveau, 2001). For example, in active voice clauses, the quantifier for the subject noun is usually preferred for wide scope over the quantifier of the predicate noun (Kurtzman and MacDonald, 1993). But such preferences can easily be overruled by world knowledge:

```
A doctor lives in every city.
```
1. $\exists_{d \in Docs} \ \forall_{c \in Cities}$ lives in$(d, c)$
   (A single doctor lives in all cities.)
2. $\forall_{c \in Cities} \ \exists_{d \in Docs}$ lives in$(d, c)$
   (Each city has a different doctor living there.)

Syntactic preferences would normally indicate that reading 1 is better, but in this particular case common-sense knowledge of the world overrules that preference and makes reading 2 far more probable.

Open-domain pragmatic knowledge is usually not available to language processing systems, but that is beginning to change. Recent research in open information extraction (Banko and Etzioni, 2008; Davidov and Rappaport, 2008) has shown that we can extract large amounts of relational data from open-domain text with high accuracy. Here, we show how we can connect the two fields, by extracting a targeted form of pragmatic knowledge for use in quantifier scope disambiguation. Our contributions are:

1) We build an extraction mechanism for extracting pragmatic knowledge about relations. In par-

ticular, we extract knowledge about the expected sizes of the sets of objects that participate in the relations. The task of identifying functional relationships is a subtask of our extraction problem that has received recent attention in the literature (Ritter et al., 2008).

2) We devise a novel probabilistic model in the Markov Logic Network framework for reasoning over possible readings of sentences that involve quantifier scope ambiguities. The model is able to assign a probability that a particular reading is plausible, given the pragmatic knowledge we extract.

3) We provide an empirical demonstration that our system is able to resolve quantifier scope ambiguities in cases where the syntactic and lexical features used by previous systems are of no help.

The remainder of this paper is organized as follows. The next section describes previous work. Section 3 shows how the problem can be formulated as a task of assigning probabilities to possible worlds, and that the crucial difference between them has to do with the number of objects participating in individual relationships. Section 4 discusses our techniques for extracting the pragmatic knowledge that allows us to make judgments about quantifier scope. Section 5 presents our probabilistic model for resolving scope ambiguities. We present an empirical study in section 6, and section 7 concludes and suggests items for future work.

## 2 Related Work

Quantifier scope disambiguation has received attention in linguistics and computational linguistics since at least the 1970s. Montague (1973) gave a seminal treatment of quantifier ambiguities, and argued that a particular syntax-based mechanism known as "quantifying-in" could resolve scope ambiguities. Since then, most work on disambiguation has focused on syntactic clues for determining which readings of an ambiguous statement are possible, and of the set of possible readings, which ones are preferred (Van Lehn, 1978; Hobbs and Shieber, 1987; Poesio, 1993a; Hurum, 1988; Moran, 1988). For instance, one linguistic study (Kurtzman and MacDonald, 1993) determined that in active voice sentences where quantifiers in the subject and object give rise to scope ambiguity, there is a preference for the reading in which the subject quantifier has wide scope — the direct reading is acceptable 70-80% of the time, whereas the indirect reading is acceptable 30-40%

of the time. Sentences that are similar in all respects except that they are passive voice have no such preference. Nevertheless, in these studies both readings are often quite plausible. In addition to syntactic clues, other studies have noted that the choice of quantifier has a significant effect on scope disambiguation (*e.g.*, "each" has a greater tendency for wide scope than "every") (Van Lehn, 1978; Alshawi, 1990). Most authors have noted that both syntactic and lexical evidence fall short of a full solution, and that pragmatic knowledge (knowledge about the world) is necessary for this task (Van Lehn, 1978; Saba and Corriveau, 1997; Moran, 1988). Saba and Corriveau (2001) recently proposed a test for quantifier scope disambiguation using pragmatic knowledge. However, they do not show how to extract the necessary information, nor do they implement or evaluate their proposed test.

Due to the difficulty of the problem, several authors have devised techniques for "underspecified" logical representations that can efficiently store multiple ambiguous readings, and they devise techniques for automated reasoning using underspecified representations (Reyle, 1995; Latecki, 1992; Poesio, 1993b). Others (Hobbs and Shieber, 1987; Park, 1988) have devised computational mechanisms for generating all of the possible readings of statements exhibiting quantifier ambiguity, especially in cases involving more than two quantifiers.

Detecting functions in extracted relational data has been studied in several contexts. Ritter *et al.*(2008) use knowledge of functions to determine when two extracted relationships contradict one another. Knowledge of functions has also been important in finding synonyms (Yates and Etzioni, 2009) and in review mining (Popescu, 2007). We extend this work by extracting not just a binary determination of whether a relation is functional, but a distribution over the expected number of arguments for that relation. Our technique also differs from previous work based on extracted relationships between named entities. We leverage domain-independent extraction patterns involving numeric phrases, as discussed below; our technique is complementary to existing approaches and could in fact be combined with them for even greater accuracy. Finally, we apply the extracted knowledge in a novel way to quantifier scope disambiguation.

Our work is similar in spirit to several recent

projects that use semantic reasoning over extracted knowledge for a novel approach to well-known tasks. For example, Schoenmackers *et al.*(2008) have recently used extracted knowledge for the task of predicting whether a new extracted fact is correct. Yates *et al.*(2006) use extracted knowledge to determine whether a parse of a sentence has a plausible semantic interpretation. We extend this new line of attack to a hard problem in language understanding.

## 3 Possible Worlds Framework

We now present a framework for reasoning about quantifier scope ambiguities, and for choosing among possible readings based on pragmatic knowledge (or world knowledge — we use the terms interchangeably). We first present a formal description of the quantifier scope disambiguation (QSD) problem. We then describe the crucial differences between the "possible worlds" evoked by different readings of an ambiguous statement.

### 3.1 Representation of Readings

We follow Copestake *et al.* (2005), among others, in representing quantifiers as modal operators with three arguments: a variable name for the variable being quantified; a logical formula, called the *restriction*, which defines the set of objects over which the variable may range; and a second logical formula, called the *body*, which defines the expression in which the quantified variable takes part. For example, we represent the sentence "Every dog barks" as: $every(x, dog(x), barks(x))$.

For the sake of clarity and convenience, we restrict our attention to a common syntactic form of sentences, where the semantic representation is relatively well-understood: active-voice English sentences in which the subject noun phrase is quantified, and a noun phrase in the predicate (either an object of the verb, or an object of a preposition attached to the verb) is also quantified. For a sentence with the following structure, in which $p_i$ and $q_j$ represent predicates introduced by modifiers like adjectives and prepositional phrases,

$$(_S \, (_{NP} \, (_{DET} \, Q_1)(_{\overline{N}} \, [p_1, \dots, p_n]C_1))$$
$$(_{VP} \, (_V \, R)(_{NP} \, (_{DET} \, Q_2)(_{\overline{N}} \, [q_1, \dots, q_m]C_2)))$$

we can represent the two possible readings of the sentence as:

**direct reading:**
$$Q_1(x, C_1(x) \wedge p_1(x) \wedge \dots \wedge p_n(x),$$
$$Q_2(y, C_2(y) \wedge q_1(y) \wedge \dots \wedge q_m(y),$$
$$R(x, y))) \tag{1}$$

**indirect reading:**
$$Q_2(y, C_2(y) \wedge q_1(y) \wedge \dots \wedge q_m(y),$$
$$Q_1(x, C_1(x) \wedge p_1(x) \wedge \dots \wedge p_n(x),$$
$$R(x, y))) \tag{2}$$

By making the restriction to this type of sentences, we can isolate the effects of pragmatics on scope disambiguation decisions from the effects of syntax, since all test cases have essentially the same syntax. As we show below, for certain types of relations, the preference for interpretations may be drastically different from the general preference for the direct reading, even though the syntax of the sentences we investigate matches the syntax studied by Kurtzman and MacDonald (1993).

### 3.2 Readings, Possible Worlds, and World Knowledge

The different logical forms for the direct and indirect readings describe different "possible worlds." For instance, the direct reading of "A doctor lives in every city" describes worlds in which there is a single doctor who manages to reside in each city of the world simultaneously. This reading is "possible" in the sense that it does not contradict itself. In logical terms, if $\phi$ represents the direct reading of this sentence, $\phi \not\vdash \bot$. Using some imagination one could devise a scenario, perhaps in an online game world, that satisfies $\phi$.

Nevertheless, the indirect reading is strongly preferred for this statement in the absence of any context that indicates an abnormal world. The indirect reading $\phi'$ describes worlds where every city is inhabited by some doctor, but potentially a different doctor per city. Using pragmatic knowledge, the reader can easily deduce that this logical statement is a much more likely reading than $\phi$. Let $B$ represent the reader's pragmatic knowledge, including facts like "People don't simultaneously live in more than one city," and, "There are at least hundreds of cities in the world." The reader can easily deduce that $B \models \neg\phi$. We now turn to methods for extracting the necessary pragmatic knowledge $B$ from text.

## 4 Extraction Techniques to Support QSD Decisions

Saba and Corriveau (2001) point out that there is a restricted form of pragmatic knowledge that can be used in many instances of QSD. Consider the facts that were used above to determine that $\phi'$ is preferable to $\phi$. The facts fall into two basic categories of knowledge: 1) the size of class $C$ (*e.g.*, how many cities are there?), and 2) the expected number of $Y$ participants in a relationship $R$, given that there is exactly 1 $X$ participant (*e.g.*, how many cities does 1 doctor live in?). In both cases, we are concerned with extracting sizes of sets.

Previous extraction systems have attempted to estimate set sizes based on extracted named entities. Downey *et al.* (2005)estimate the size of classes based on the number of named-entities extracted for the class. As far as we are aware, finding the expected size of an argument set for a relation is a novel task for information extraction, but several researchers (Ritter et al., 2008; Yates and Etzioni, 2009) have investigated the special case of detecting functional relations — those relations where the expected size of the $Y$ argument set is precisely 1. As with class size extraction, they use extractions involving named-entity arguments to find functional relations.

Approaches that depend on named-entity extractions have several disadvantages: they must find a large set of named-entities for every set, which can be time-consuming and difficult. Also, many classes, like "trees" and "hot dogs," have no or very few named instances, but many un-named instances, so approaches based on named entities have little hope. In fact, besides classes like people, locations, and organizations (and their subclasses), there are few classes that have a large number of named instances. For classes that do have named instances, synonymy, polysemy, and extraction errors are common problems that can all affect estimates of size (Ritter et al., 2008).

Rather than indirectly determining set sizes from extracted instances, our system directly extracts estimates of set sizes. It uses *numeric phrases*, like "two trees," "hundreds of students," or "billions of stars," to associate numeric values with sets. Table 1 lists the numeric phrases we use. Currently, we use only numeric phrases with explicit values or ranges of values, but it may be possible to increase the recall of our extraction technique by incorporating more approximate phrases

| Numeric Phrase | Value |
| --- | --- |
| no \| none \| zero | 0 |
| a \| one \| this \| the | 1 |
| two | 2 |
| ⋮ | ⋮ |
| one hundred \| a hundred | 100 |
| ⋮ | ⋮ |
| hundreds of | 100 |
| thousands of | 1,000 |
| tens of thousands of | 10,000 |
| ⋮ | ⋮ |

Table 1: **Numeric phrases used in our extraction patterns.** For the word "the", we require that it be followed directly by a singular noun, to try to weed out plural usages.

like "several," "many," or even bare plurals. We do not match numbers expressed in digits (*e.g.*, 1234) because we found that they produced too many noisy extractions, such as dates and times. For words like "hundreds," we set the value of the word to be the lower limit (*i.e.*, 100). This gives a conservative estimate of the value, but our techniques described below can help to compensate for this bias.

Table 2 lists examples of the hand-crafted, domain-independent extraction patterns we use. Our extraction patterns generate two types of extractions, one for classes and one for relations. For classes, each extraction $E$ consists of a class name $E_c$ and a number $E_n$ indicating the size of some subset $S \subseteq E_c$. For instance, the 4gram "hundreds of students are" matches our first pattern. The numeric phrase "hundreds of" here indicates that some subset $S \subseteq E_c =$ students has a size in the hundreds. After processing a large corpus, our system can determine a probability distribution for the size of a class given by:

$$P_C(\texttt{size}(C) = N) = \frac{|\{E \mid E_c = C \wedge E_n = N\}|}{|\{E \mid E_c = C\}|}$$

In practice, we only include the largest 20% of the numbers $N$ in the set of extractions for a class to estimate that class's size.

The second type of extraction we get from our patterns are relational extractions. Each relational extraction $F$ consists of a relation name $F_r$, and possibly names for the classes of its two arguments, $F_{c1}, F_{c2}$. In addition, the extraction contains values for the size of both arguments, $F_{n1}$

| Pattern | Extraction |
|---|---|
| `<numeric> <word>+(of \| are \| have)` | $E_c = \text{<word>+}, E_n = value(\text{<numeric>})$ |
| `(I \| he \| she) <word>+ <numeric> <noun>` | $F_r = \text{<word>+}, F_{c1} = \text{people}, F_{c2} = \text{<noun>},$ $F_{n1} = 1, F_{n2} = value(\text{<numeric>})$ |
| `it is` *pastParticiple*`(<verb>) by <numeric>` | $F_r = \text{<verb>}, F_{c2} = \text{thing},$ $F_{n1} = value(\text{<numeric>}), F_{n2} = 1$ |
| `is the <word> of <numeric>` | $F_r = \text{is the <word> of},$ $F_{n1} = 1, F_{n2} = value(\text{<numeric>})$ |

Table 2: **Sample extraction patterns for discovering classes ($E_c$) and their sizes ($E_n$), or relations ($F_r$) and the expected set size of their arguments ($F_{n1}$ and $F_{n2}$).**

and $F_{n2}$ respectively. For example, the fragment "she visited four countries" matches the second pattern in Table 2, with $F_r =$ visited, $F_{n1} = 1$, and $F_{n2} = 4$. Note that in our extraction patterns, one of of the arguments is always constrained to be a singleton set, like "he" or "it." This restriction allows us to avoid quantifier scope ambiguity in the extraction process: if we extracted phrases like "Two men married two women," it would be unclear which quantifier has wide scope, and therefore how many men and women are participating in each "married" relationship. By using singular pronouns, we avoid this confusion; in almost all cases, these pronouns have wide scope, and indicate a single element.[1]

Based on these extractions, our system determines two distributions for each relation: $P_R^{Left}(n)$ and $P_R^{Right}(n)$. The $P_R^{Left}$ distribution represents the probability that the left argument of $R$ is a set of size $n$, given that the right argument is a singleton set, and likewise for $P_R^{Right}$. We determine the distributions from the extractions by maximum likelihood estimation:

$$P_R^{Left}(n) = \frac{|\{F \mid F_r = R, F_{n1} = n, F_{n2} = 1\}|}{|\{F \mid F_r = R, F_{n2} = 1\}|}$$

$$P_R^{Right}(n) = \frac{|\{F \mid F_r = R, F_{n2} = n, F_{n1} = 1\}|}{|\{F \mid F_r = R, F_{n1} = 1\}|}$$

For example, for the relation `is the father of`, we might see the fragment "he is the father of two children" far more often than "he is the father of twenty children." $P_{\text{is the father of}}^{Right}$ would therefore have a relatively low probability for $n = 20$. As one would expect, the relation

---
[1] An example of an exception to this rule from our data set is the sentence "It is worn by millions of women." Here, "it" refers to a class of items such as a brand, and thus may refer to a different item for each of the "millions of women."

`visited` appears more often with "twenty," and the relation `married` never does. Their $P^{Right}$ distributions are comparatively higher and lower, respectively, than the one for `is the father of` at $n = 20$.

In practice, we create histograms of the extracted counts for both our $E$ and $F$ extractions, and our probability distributions are really distributions over the buckets in these histograms, rather than over all possible set sizes. To help combat sparse counts for large numeric values, we use buckets of exponentially increasing width for larger numeric values. Thus between $n = 0$ and 10, buckets have size 1, between 10 and 100 they have size 10, and so on.

We also create distributions in the same way for relations together with their extracted argument classes. Since counts for these extractions tend to be much more sparse, we interpolate these distributions with the distribution for just the relation, and with the distribution for the relation and just one class. We use equal weights for all interpolated distributions.

## 5 A Probabilistic Model for Quantifier Scope Disambiguation

QSD requires reasoning about different possible states of the world. This involves logical reasoning, since the direct and indirect readings differ in the number of objects that exist in models satisfying each reading, and the number of relationships between those objects. QSD also involves probabilistic reasoning, since none of the extracted knowledge is certain. We leverage recent work on Markov Logic Networks (MLNs) (Richardson and Domingos, 2006) to incorporate both types of reasoning into our technique for QSD. We next briefly review MLNs, before describing our model and

methods for training it.

## 5.1 Markov Logic Networks

Syntactically, an MLN consists of a set of first-order logical formulas $\mathbf{F}$ and a real-valued weight $w_F$ for each $F \in \mathbf{F}$. Semantically, an MLN defines a probability distribution over possible *groundings* of the logical formulas. That is, if $U$ denotes the set of all objects in the universe, and $\mathbf{G}$ denotes the set of all possible ways to ground every $F \in \mathbf{F}$ (*i.e.*, substitute an element from $U$ for every variable in $F$), then an MLN defines a distribution over truth assignments to the grounded formulas $G \in \mathbf{G}$. Let $\mathcal{I}$ denote the set of all possible interpretations of $\mathbf{G}$ — that is, each $I \in \mathcal{I}$ assigns true or false to every $G \in \mathbf{G}$. The probability of a particular interpretation $I$ according to the MLN is given by :

$$
\begin{aligned}
P(I) &= \frac{1}{Z} \exp\left( \sum_{F \in \mathbf{F}} w_F \cdot n(F, I) \right) \\
Z &= \sum_{I \in \mathcal{I}} \exp\left( \sum_{F \in \mathbf{F}} w_F \cdot n(F, I) \right)
\end{aligned}
$$

where $n(F, I)$ gives the number of groundings of $F$ that are true in interpretation $I$.

The equation above provides an expression for $P(I)$ when $U$, or at least the size of $U$, is known and fixed. When we are interpreting expressions like "every city" or "every doctor", however, we require extracted knowledge to inform the system of the correct number of "city" or "doctor" objects. Since our extractions are uncertain, they provide a distribution $P(|U| = n)$ for the size of a class. Using $P(|U|)$, we can still calculate $P(I)$, even without knowing the exact size of $U$:

$$
P(I) = \sum_n P(|U| = n) P(I \mid |U| = n)
$$

## 5.2 MLN Classifier for QSD

Let $Q$ be a QSD problem, consisting of a relation $Q_r$, a class for the first argument of the relation $Q_{c1}$, a class for the second argument $Q_{c2}$, and quantifiers $Q_{q1}$, $Q_{q2}$ for each argument. We construct an MLN model for $Q$ using the following logical formulas:

1) *Clustering:* We allow members of each class to belong to clusters denoted by $\gamma$, but each element can belong to no more than one cluster. This is represented by the following formula, which has

infinite weight.

$$
\forall_{x \in Q_{c1} \cup Q_{c2}, \gamma, \gamma'} x \in \gamma \wedge x \in \gamma' \Rightarrow \gamma = \gamma'
$$

2) *Relation between clusters:* Every cluster of class 1 elements must participate in the relation $Q_r$ with exactly one cluster of class 2 elements, and *vice versa*. We represent this participation in $Q_r$ with a series of logical relations $R_{m,n}$, each of which indicates that a cluster of size $m$ is participating in $Q_r$ with a cluster of size $n$. We use a set of formulas for each setting of $m$ and $n$, each having infinite weight.

$$
\begin{aligned}
&\forall_{\gamma \subset Q_{c1}} \exists!_{\gamma' \subset Q_{c2}, m, n} R_{m,n}(\gamma, \gamma') \\
&\forall_{\gamma' \subset Q_{c2}} \exists!_{\gamma \subset Q_{c1}, m, n} R_{m,n}(\gamma, \gamma') \\
&\forall_{\gamma, \gamma'} R_{m,n}(\gamma, \gamma') \Rightarrow |\gamma| = m \wedge |\gamma'| = n
\end{aligned}
$$

3) *Prefer relations between clusters of the appropriate size:* We include a set of formulas with finite weight that express the preference for a particular relation to have arguments of a certain size. There is a separate formula for each setting of $m$ and $n$, with a separate weight $w_{m,n}$ for each.

$$
\forall_{\gamma, \gamma'} R_{m,n}(\gamma, \gamma')
$$

This formula does most of the work of our classifier. For a given relation, such as the lives in(Person, City) relation, we can set the weights $w_{m,n}$ so that the model prefers worlds where each person lives in just one place. For instance, we can set the weight $w_{1,1}$ relatively high, so that the model is more likely to make clusters of size 1, which then participate in the $R_{1,1}$ relation.

We describe how we choose the $w_{m,n}$ weights below, but first we explain how to incorporate the quantifiers $Q_{q1}$ and $Q_{q2}$ into the model. Unfortunately, every natural language quantifier has different semantics (Barwise and Cooper, 1981), and thus they affect our model in different ways. Here, we restrict our attention to the two common quantifiers "a" and "every", but note that the MLN framework is a powerful tool for incorporating the logical semantics and statistical preferences of other quantifiers.

For the quantifier "a", we require that the relation have no argument clusters with size more than 1 for that class. Thus if $Q_{q1} = $ "a", we restrict $R_{m,n}$ to $R_{1,n}$, and *vice versa* if $Q_{q2} = $ "a." Furthermore, we require that at least one element of the class belong to a cluster: $\exists_{x, \gamma} x \in \gamma$ has infinite weight. For "every," we require that every element of the class that "every" modifies to be part

of some cluster. To effect this change, we simply put an infinite weight on the formula $\forall_x \exists_\gamma x \in \gamma$.

Our MLN model is general in the sense that for any QSD problem $Q$, it can determine probabilities for any possible world corresponding to a reading of $Q$. For our purposes, we are primarily interested in the direct and indirect readings of any $Q$ involving "a" and "every." To predict the correct reading for a given $Q$, we simply check to see which has the higher probability according to our MLN model.

### 5.3 Parameter Estimation

Our MLN model for QSD requires settings for the $w_{m,n}$ parameters for each QSD problem $Q$. The standard approach to this problem would be to estimate these parameters from labeled training data. We reject the standard supervised framework, however, because each distinct relation $Q_r$ requires different settings of the parameters, and therefore a standard supervised approach would require manually labeled training data for every relation $Q_r$.

A second approach that is made possible by our extraction technique is to set the parameters using the extracted distributions. We tried this approach by setting $w_{1,n} = \log P_{Q_r}^{Right}(n)$ and $w_{m,1} = \log P_{Q_r}^{Left}(m)$; since we only consider sentences containing the quantifier "a", one of $m$ and $n$ will always be 1. Unfortunately, in our experiments we found that this setting for the parameters often gave far too little weight for large values of $m$ and $n$, and as a consequence, the classifier would systematically judge one reading to be more likely than another.

To counteract this problem, we take a hybrid approach to parameter estimation, informed by both labeled training data and the extracted distributions. Crucially, our approach, which we call ZIPF FLATTENING, has only two parameters that need to be trained using a supervised approach, and these parameters do not depend on the relation $R$. Thus, the approach minimizes the amount of training data we need to a practical level.

ZIPF FLATTENING works by correcting the $P_R$ distributions to give higher weight to larger values of $m$ and $n$. First, we estimate a Zipf distribution from the raw extracted counts for each argument of relation $R$. To fit a Zipf curve, we use least-squares linear regression on the log-log plot of the extracted counts to find parameters $z_R$ and $c_R$ such

that

$$
\begin{aligned}
\log\left(count\right) &= z_R \cdot \log\left(argSize\right) + c_R \\
\Rightarrow count &= e^{c_R} \cdot argSize^{z_R}
\end{aligned}
$$

We can perform this part automatically, using only the extraction data and no manually labeled training data, for every relation. However, the fitted Zipf distribution needs to be corrected for the systematic bias in the extracted counts. To do this, we introduce two parameters, $\alpha_1$ and $\alpha_2$, that we use to scale back the sharp falloff in the Zipf distribution. Our *flattened* distribution has the form:

$$
count = e^{\alpha_1 c_R} \cdot argSize^{\alpha_2 z_R}
$$

When $\alpha_2$ is less than 1, the resulting curve has a less steep slope, and greater weight is placed on the large values of $m$ and $n$, as desired. Our last step is to interpolate the $P_R^{Right}$ and $P_R^{Left}$ distributions with the flattened Zipf distribution to come up with corrected distributions for the right and left argument sizes of $R$. We use equal weights on the two distributions to interpolate. Note that if the original counts from the extraction system include counts for only one argument size, then it is impossible to estimate a Zipf distribution, and we simply fall back on the extracted distribution. We do not include counts for an argument size of zero in this process.

To estimate the parameters $\alpha_i$, we collect a training set of QSD problems $Q$, labeled with the correct reading for each (direct or indirect), and run the extractor for the relations $Q_r$ appearing in the training set. We then perform a gradient descent search to find optimal settings for the $\alpha_i$ on the training data.

## 6 Experiments

We report on two sets of experiments. The first tests our extraction technique on its own, and the second tests the accuracy of our complete QSD system, including the extraction mechanisms and the prediction model, on a quantifier scope disambiguation task.

### 6.1 Function Detection Experiment

Function detection is an important task in its own right, and has been used in several previous applications (Ritter et al., 2008; Yates and Etzioni, 2009; Popescu, 2007). To turn our extraction system into a classifier for functions vs. non-functions, we simply checked whether there were

|              | Num | Precision | Recall | F1 |
|--------------|-----|-----------|--------|-----|
| Functions    | 54  | .79       | .76    | .77 |
| Non-functions| 74  | .83       | .85    | .84 |

Table 3: **Precision and recall for detecting functions using the numeric extraction technique.**

any extractions for $R$ with $F_{n2} > 1$. If so, we predicted that the $R$ was nonfunctional, and otherwise we predicted it was functional.

We used the Web1Tgram Corpus of n-grams provided by Google, Inc to extract classes, relations, and counts. This corpus contains counts for 2- through 5-grams that appear on the Web pages indexed by Google. Counts are included in this data set for all n-grams that appeared at least 40 times in their text. We ran our extraction techniques on the 3-, 4- and 5-grams. To create a test set, we sampled a set of 200 relations from our extractions, removed any relations that consisted of punctuations, stopwords, or other non-relational items. We then manually labeled the remainder as functions or non-functions.

Table 3 shows our results. A baseline system that simply predicts the majority class (non-functions) on this data set would achieve an accuracy of 56%, well below the 81% accuracy of our classifier. Many of the relations in our test set, like `built(Person, House)` and `is riding(Person,Animal)`, do not ordinarily have named-entity extractions for both arguments, and would therefore not be amenable to previous function detection approaches.

Some of our technique's errors highlight interesting difficulties with function detection. For instance, while we labeled the `is capital of` relation as a function, our technique predicted that it was not. It turns out that the country of Bolivia has two capitals, and the South Asian region of Jammu and Kashmir also has two capitals. Both of these facts are prominent enough on the Web to cause our system to detect a small probability for $P^{Right}_{capital\ of}(2)$. Thus any label for this relation is somewhat unsatisfying: it is almost entirely functional, but not strictly so. By generalizing the problem to one of determining a distribution for the size of the argument, we can handle these border cases in a useful way for QSD, as discussed below.

## 6.2 Preliminary QSD Experiments

We test our complete QSD system on two important tasks. In the first, the system is presented with a series of QSD problems $Q$ in which the first quantifier $Q_{q1}$ is always "a," and the second ($Q_{q2}$) is always "every." Each example is manually labeled to indicate whether a direct or indirect reading of the sentence is preferred, and the system is charged with predicting the preferred reading. In the second task, each $Q$ has "every" as the first quantifier, and "a" as the second quantifier. Since indirect readings are very rarely preferred for active-voice sentences of this form,we charge the system with making a different type of prediction: determine whether the indirect reading is plausible or not. The system assumes that every sentence has a plausible direct reading, but by determining whether the indirect reading is plausible, it can determine whether the sentence is ambiguous between the two readings.

We created data sets for these tasks by sampling our 5grams for examples containing the relations in our function experiment. From this set, we selected phrases that involved named classes for the arguments to the relation. When a class was missing, we either manually supplied one, or discarded the example. We then constructed two examples from each combination of relation and argument classes: one example in which the first argument is constrained by the quantifier "a" and the second by "every," and a second example in which the quantifiers are reversed. Finally, we manually labeled every example with a preference for direct or indirect reading (in the case of "a/every" examples) or with a plausibility judgment for the indirect reading (in the case of "every/a" examples). Our final test sets included 46 labeled examples for each task. Further experiments involving multiple annotators, as in the experiments of Kurtzman and MacDonald (1993), are of course desirable, but note that even their experiments included just 32 labeled examples.

Table 4 shows our results for the first QSD task, and Table 5 shows our results for the second one. In each case, we compare our supervised Corrected MLN model against an Uncorrected MLN model that uses no supervised data, and simply takes its weights straight from our extracted distributions. The supervised model uses a training corpus of 10 manually labeled examples for each task, five from each class. We also compare against a majority class baseline. Note that the Corrected

1472

| System | Acc. | Direct | | Indirect | |
|---|---|---|---|---|---|
| | | P | R | P | R |
| All-Direct BL | .53 | .53 | 1.0 | 0.0 | 0.0 |
| Uncorrected MLN | .58 | .78 | .30 | .53 | .90 |
| Corrected MLN | **.74** | .77 | .74 | .71 | .75 |

Table 4: **Our trained MLN outperforms two other systems at predicting whether sentences of the form "A/some <class 1> <relation> every <class 2>" should have direct or indirect readings.** We measure accuracy over the whole dataset, as well as precision and recall for the two subsets labeled with direct and indirect readings, respectively.

| System | Acc. | Plausible | | Implaus. | |
|---|---|---|---|---|---|
| | | P | R | P | R |
| All-Plausible BL | .67 | .67 | 1.0 | 0.0 | 0.0 |
| Uncorrected MLN | .49 | .89 | .28 | .38 | .93 |
| Corrected MLN | **.72** | .76 | .86 | .60 | .43 |

Table 5: **Our trained MLN outperforms two other systems at predicting whether sentences of the form "Every <class 1> <relation> a/some <class 2>" have a plausible indirect reading or not.** We measure accuracy over the whole dataset, as well as precision and recall for the two subsets labeled with plausible and implausible indirect readings.

MLN model has balanced recall numbers for the two classes in both of our tasks, compared with the Uncorrected MLN. This indicates that our ZIPF FLATTENING technique is accurately learning better weights to remove the systematic bias in the Uncorrected MLN.

Our results demonstrate the utility of our extracted distributions for these difficult tasks. Although the extracted data prevents us from determining that `is capital of` should be classified as a function, since almost all of the probability mass in $P^{Right}$ is still on $n \in \{0, 1\}$. Thus, the probability for the direct reading of a sentence like "Some city is the capital of every country" is still very low. Likewise, even though our system (correctly) determines that the relation `is a parent of` is non-functional, it does not therefore group it with other non-functional relations like `visited`. The distribution $P^{Right}_{\text{is parent of}}(n)$ is skewed to much smaller numbers for $n$ than is the distribution for `visited`, and thus the indirect reading for "A person is the parent of every child" is much more likely than the indirect reading of "A person visited every country."

The biggest hurdle for better performance is noise in our extraction technique. Polysemous relations sometimes have large counts for large argument sizes in one sense, but not another. Using argument classes to disambiguate relations can help, but extractions for relations in combination with argument classes are much more sparse. Improved extraction techniques could directly impact performance on the QSD task.

## 7 Conclusion and Future Work

We have demonstrated targeted methods for extracting world knowledge that is necessary for making quantifier scope disambiguation decisions. We have also demonstrated a novel, minimally-supervised, statistical relational model in the Markov Logic Network framework for making QSD decisions based on extracted pragmatics.

While our preliminary results for QSD are promising, there are clearly many areas for improvement. We will need to handle more kinds of quantifiers in our MLN model. Our current system is biased towards using purely pragmatic knowledge, but a complete system should also integrate syntactic and lexical constraints and preferences. Also, discourses can introduce knowledge that directly affects QSD problems, such as constraints on the size of a particular set that is discussed in the discourse. Integrating our technique for QSD with discourse processing is a major challenge that we hope to address.

## References

Hiyan Alshawi. 1990. Resolving quasi logical forms. *Computational Linguistics*, 16(3):133–144.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional information extraction. In *Proceedings of the ACL*.

J. Barwise and R. Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):150–219.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3:281–332.

D. Davidov and A. Rappaport. 2008. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated SAT analogy questions. In *Proceedings of the ACL*.

Doug Downey, Oren Etzioni, and Stephen Soderland. 2005. A Probabilistic Model of Redundancy in Information Extraction. In *IJCAI*.

Jerry R. Hobbs and Stuart M. Shieber. 1987. An algorithm for generating quantifier scopings. *Computational Linguistics*, 13(1-2):47–63.

Sven Hurum. 1988. Handling scope ambiguities in English. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 58–65.

Howard S. Kurtzman and Maryellen C. MacDonald. 1993. Resolution of quantifier scope ambiguities. *Cognition*, 48:243–279.

Longin Latecki. 1992. Connection relations and quantifier scope. In *Proceedings of the ACL*.

Richard Montague. 1973. The proper treatment of quantification in ordinary English. In Jaakko Hintikka, Julius Moravcsik, and Patrick Suppes, editors, *Approaches to Natural Languages*, pages 221–242. Reidel, Dordrecht.

Douglas B. Moran. 1988. Quantifier scoping in the SRI core language engine. In *Proceedings of the 26th Annual Meeting of the Assoc. for Comp. Linguistics*, pages 33–40.

Jong C. Park. 1988. Quantifier scope and constituency. In *Proceedings of the 26th Annual Meeting of the Assoc. for Comp. Linguistics*, pages 33–40.

Massimo Poesio. 1993a. Assigning a semantic scope to operators. In *Proceedings of the ACL*.

Massimo Poesio. 1993b. Assigning a semantic scope to operators. In *Proceedings of the Second Conference on Situation Theory and Its Applications*.

Ana-Maria Popescu. 2007. *Information Extraction from Unstructured Web Text*. Ph.D. thesis, University of Washington.

Uwe Reyle. 1995. On reasoning with ambiguities. In *Proceedings of the EACL*, pages 1–8.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62:107–136.

Alan Ritter, Doug Downey, Stephen Soderland, and Oren Etzioni. 2008. It's a contradiction — No, it's not: A case study using functional relations. In *Empirical Methods in Natural Language Processing*.

Walid S. Saba and Jean-Pierre Corriveau. 1997. A pragmatic treatment of quantification in natural language. In *Proceedings of the National Conference on Artificial Intelligence*.

Walid S. Saba and Jean-Pierre Corriveau. 2001. Plausible reasoning and the resolution of quantifier scope ambiguities. *Studia Logica*, 67:271–289.

Stefan Schoenmackers, Oren Etzioni, and Dan Weld. 2008. Scaling textual inference to the web. In *Proceedings of EMNLP*.

Kurt Van Lehn. 1978. Determining the scope of English quantifiers. Technical Report AI-TR-483, AI Lab, MIT.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research (JAIR)*, 34:255–296, March.

Alexander Yates, Stefan Schoenmackers, and Oren Etzioni. 2006. Detecting parser errors using web-based semantic filters. In *Proceedings of EMNLP*.