Learning to Extract Relations from the Web using Minimal Supervision

Razvan C. Bunescu
Department of Computer Sciences
University of Texas at Austin
1 University Station C0500
Austin, TX 78712
razvan@cs.utexas.edu

Raymond J. Mooney
Department of Computer Sciences
University of Texas at Austin
1 University Station C0500
Austin, TX 78712
mooney@cs.utexas.edu

Abstract

We present a new approach to relation extraction that requires only a handful of training examples. Given a few pairs of named entities known to exhibit or not exhibit a particular relation, bags of sentences containing the pairs are extracted from the web. We extend an existing relation extraction method to handle this weaker form of supervision, and present experimental results demonstrating that our approach can reliably extract relations from web documents.

1 Introduction

A growing body of recent work in information extraction has addressed the problem of *relation extraction* (RE), identifying relationships between entities stated in text, such as `LivesIn(Person, Location)` or `EmployedBy(Person, Company)`. Supervised learning has been shown to be effective for RE (Zelenko et al., 2003; Culotta and Sorensen, 2004; Bunescu and Mooney, 2006); however, annotating large corpora with examples of the relations to be extracted is expensive and tedious.

In this paper, we introduce a supervised learning approach to RE that requires only a handful of training examples and uses the web as a corpus. Given a few pairs of well-known entities that clearly exhibit or do not exhibit a particular relation, such as `CorpAcquired`(Google, YouTube) and `not`(`CorpAcquired`(Yahoo, Microsoft)), a search engine is used to find sentences on the web that mention both of the entities in each of the pairs. Although not all of the sentences for positive pairs will state the desired relationship, many of them will. Presumably, none of the sentences for negative pairs state the targeted relation. *Multiple instance learning* (MIL) is a machine learning framework that exploits this sort of weak supervision, in which a *positive bag* is a set of instances which is guaranteed to contain at least one positive example, and a *negative bag* is a set of instances all of which are negative. MIL was originally introduced to solve a problem in biochemistry (Dietterich et al., 1997); however, it has since been applied to problems in other areas such as classifying image regions in computer vision (Zhang et al., 2002), and text categorization (Andrews et al., 2003; Ray and Craven, 2005).

We have extended an existing approach to relation extraction using support vector machines and string kernels (Bunescu and Mooney, 2006) to handle this weaker form of MIL supervision. This approach can sometimes be misled by textual features correlated with the specific entities in the few training pairs provided. Therefore, we also describe a method for weighting features in order to focus on those correlated with the target relation rather than with the individual entities. We present experimental results demonstrating that our approach is able to accurately extract relations from the web by learning from such weak supervision.

2 Problem Description

We address the task of learning a relation extraction system targeted to a fixed binary relationship R. The only supervision given to the learning algo-

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 576–583,
Prague, Czech Republic, June 2007.

rithm is a small set of pairs of named entities that are known to belong (positive) or not belong (negative) to the given relationship. Table 1 shows four positive and two negative example pairs for the corporate acquisition relationship. For each pair, a bag of sentences containing the two arguments can be extracted from a corpus of text documents. The corpus is assumed to be sufficiently large and diverse such that, if the pair is positive, it is highly likely that the corresponding bag contains at least one sentence that explicitly asserts the relationship R between the two arguments. In Section 6 we describe a method for extracting bags of relevant sentences from the web.

$+/-$	**Arg** a_1	**Arg** a_2
$+$	Google	YouTube
$+$	Adobe Systems	Macromedia
$+$	Viacom	DreamWorks
$+$	Novartis	Eon Labs
$-$	Yahoo	Microsoft
$-$	Pfizer	Teva

Table 1: Corporate Acquisition Pairs.

Using a limited set of entity pairs (e.g. those in Table 1) and their associated bags as training data, the aim is to induce a relation extraction system that can reliably decide whether two entities mentioned in the same sentence exhibit the target relationship or not. In particular, when tested on the example sentences from Figure 1, the system should classify S_1, S_3,and S_4 as positive, and S_2 and S_5 as negative.

$+/S_1$: Search engine giant **Google** has bought video-sharing website **YouTube** in a controversial $1.6 billion deal.

$-/S_2$: The companies will merge **Google**'s search expertise with **YouTube**'s video expertise, pushing what executives believe is a hot emerging market of video offered over the Internet.

$+/S_3$: **Google** has acquired social media company, **YouTube** for $1.65 billion in a stock-for-stock transaction as announced by Google Inc. on October 9, 2006.

$+/S_4$: Drug giant **Pfizer Inc.** has reached an agreement to buy the private biotechnology firm **Rinat Neuroscience Corp.**, the companies announced Thursday.

$-/S_5$: He has also received consulting fees from Alpharma, Eli Lilly and Company, **Pfizer**, Wyeth Pharmaceuticals, **Rinat Neuroscience**, Elan Pharmaceuticals, and Forest Laboratories.

Figure 1: Sentence examples.

As formulated above, the learning task can be seen as an instance of *multiple instance learning*. However, there are important properties that set it apart from problems previously considered in MIL. The most distinguishing characteristic is that the number of bags is very small, while the average size of the bags is very large.

3 Multiple Instance Learning

Since its introduction by Dietterich (1997), an extensive and quite diverse set of methods have been proposed for solving the MIL problem. For the task of relation extraction, we consider only MIL methods where the decision function can be expressed in terms of kernels computed between bag instances. This choice was motivated by the comparatively high accuracy obtained by kernel-based SVMs when applied to various natural language tasks, and in particular to relation extraction. Through the use of kernels, SVMs (Vapnik, 1998; Schölkopf and Smola, 2002) can work efficiently with instances that implicitly belong to a high dimensional feature space. When used for classification, the decision function computed by the learning algorithm is equivalent to a hyperplane in this feature space. Overfitting is avoided in the SVM formulation by requiring that positive and negative training instances be maximally separated by the decision hyperplane.

Gartner *et al.* (2002) adapted SVMs to the MIL setting using various multi-instance kernels. Two of these – the normalized set kernel, and the statistic kernel – have been experimentally compared to other methods by Ray and Craven (2005), with competitive results. Alternatively, a simple approach to MIL is to transform it into a standard supervised learning problem by labeling all instances from positive bags as positive. An interesting outcome of the study conducted by Ray and Craven (2005) was that, despite the class noise in the resulting positive examples, such a simple approach often obtains competitive results when compared against other more sophisticated MIL methods.

We believe that an MIL method based on multi-instance kernels is not appropriate for training datasets that contain just a few, very large bags. In a multi-instance kernel approach, only bags (and not instances) are considered as training examples, which means that the number of support vectors is going to be upper bounded by the number of training bags. Taking the bags from Table 1 as a sample training set, the decision function is going to be specified by at most seven parameters: the coefficients for at most six support vectors, plus an optional bias parameter. A hypothesis space characterized by such a small number of parameters is likely to have insufficient capacity.

Based on these observations, we decided to transform the MIL problem into a standard supervised problem as described above. The use of this approach is further motivated by its simplicity and its observed competitive performance on very diverse datasets (Ray and Craven, 2005). Let $\mathcal{X}$ be the set of bags used for training, $\mathcal{X}_p \subseteq \mathcal{X}$ the set of positive bags, and $\mathcal{X}_n \subseteq \mathcal{X}$ the set of negative bags. For any instance $x \in X$ from a bag $X \in \mathcal{X}$, let $\phi(x)$ be the (implicit) feature vector representation of x. Then the corresponding SVM optimization problem can be formulated as in Figure 2:

minimize:

$$\mathbf{J}(w, b, \xi) = \frac{1}{2}\|w\|^2 + \frac{C}{L}\left(c_p \frac{L_n}{L}\Xi_p + c_n \frac{L_p}{L}\Xi_n\right)$$

$$\Xi_p = \sum_{X \in \mathcal{X}_p} \sum_{x \in X} \xi_x$$

$$\Xi_n = \sum_{X \in \mathcal{X}_n} \sum_{x \in X} \xi_x$$

subject to:

$$w\,\phi(x) + b \geq +1 - \xi_x, \;\; \forall x \in X \in \mathcal{X}_p$$

$$w\,\phi(x) + b \leq -1 + \xi_x, \;\; \forall x \in X \in \mathcal{X}_n$$

$$\xi_x \geq 0$$

Figure 2: SVM Optimization Problem.

The capacity control parameter C is normalized by the total number of instances $L = L_p + L_n = \sum_{X \in \mathcal{X}_p} |X| + \sum_{X \in \mathcal{X}_n} |X|$, so that it remains independent of the size of the dataset. The additional non-negative parameter c_p ($c_n = 1 - c_p$) controls the relative influence that false negative vs. false positive errors have on the value of the objective function. Because not all instances from positive bags are real positive instances, it makes sense to have false negative errors be penalized less than false positive errors (i.e. $c_p < 0.5$).

In the dual formulation of the optimization problem from Figure 2, bag instances appear only inside dot products of the form $K(x_1, x_2) = \phi(x_1)\phi(x_2)$. The kernel K is instantiated to a subsequence kernel, as described in the next section.

4 Relation Extraction Kernel

The training bags consist of sentences extracted from online documents, using the methodology described in Section 6. Parsing web documents in order to obtain a syntactic analysis often gives unreliable results – the type of narrative can vary greatly from one web document to another, and sentences with grammatical errors are frequent. Therefore, for the initial experiments, we used a modified version of the subsequence kernel of Bunescu and Mooney (2006), which does not require syntactic information. This kernel computes the number of common subsequences of tokens between two sentences. The subsequences are constrained to be "anchored" at the two entity names, and there is a maximum number of tokens that can appear in a sequence. For example, a subsequence feature for the sentence S_1 in Figure 1 is $\tilde{s}$ = "$\langle e_1 \rangle$ ... *bought* ... $\langle e_2 \rangle$ ... *in* ... *billion* ... *deal*", where $\langle e_1 \rangle$ and $\langle e_2 \rangle$ are generic placeholders for the two entity names. The subsequence kernel induces a feature space where each dimension corresponds to a sequence of words. Any such sequence that matches a subsequence of words in a sentence example is down-weighted as a function of the total length of the gaps between every two consecutive words. More exactly, let $s = w_1 w_2 \ldots w_k$ be a sequence of k words, and $\tilde{s} = w_1\, g_1\, w_2\, g_2\, \ldots\, w_{k-1}\, g_{k-1}\, w_k$ a matching subsequence in a relation example, where g_i stands for any sequence of words between w_i and w_{i+1}. Then the sequence s will be represented in the relation example as a feature with weight computed as $\tau(s) = \lambda^{g(\tilde{s})}$. The parameter λ controls the magnitude of the gap penalty, where $g(\tilde{s}) = \sum_i |g_i|$ is the total gap.

Many relations, like the ones that we explore in the experimental evaluation, cannot be expressed without using at least one content word. We therefore modified the kernel computation to optionally ignore subsequence patterns formed exclusively of stop words and punctuation signs. In Section 5.1, we introduce a new weighting scheme, wherein a weight is assigned to every token. Correspondingly, every sequence feature will have an additional multiplicative weight, computed as the product of the weights of all the tokens in the sequence. The aim of this new weighting scheme, as detailed in the next section, is to eliminate the bias caused by the special structure of the relation extraction MIL problem.

5 Two Types of Bias

As already hinted at the end of Section 2, there is one important property that distinguishes the current MIL setting for relation extraction from other MIL problems: the training dataset contains very few bags, and each bag can be very large. Consequently, an application of the learning model described in Sections 3 & 4 is bound to be affected by the following two types of bias:

▪ **[Type I Bias]** By definition, all sentences inside a bag are constrained to contain the same two arguments. Words that are semantically correlated with either of the two arguments are likely to occur in many sentences. For example, consider the sentences S_1 and S_2 from the bag associated with "Google" and "YouTube" (as shown in Figure 1). They both contain the words "search" – highly correlated with "Google", and "video" – highly correlated with "YouTube", and it is likely that a significant percentage of sentences in this bag contain one of the two words (or both). The two entities can be mentioned in the same sentence for reasons other than the target relation R, and these noisy training sentences are likely to contain words that are correlated with the two entities, without any relationship to R. A learning model where the features are based on words, or word sequences, is going to give too much weight to words or combinations of words that are correlated with either of individual arguments. This overweighting will adversely affect extraction performance through an increased number of errors. A method for eliminating this type of bias is introduced in Section 5.1.

▪ **[Type II Bias]** While Type I bias is due to words that are correlated with the arguments of a relation instance, the Type II bias is caused by words that are specific to the relation instance itself. Using FrameNet terminology (Baker et al., 1998), these correspond to instantiated frame elements. For example, the corporate acquisition frame can be seen as a subtype of the "Getting" frame in FrameNet. The *core* elements in this frame are the *Recipient* (e.g. Google) and the *Theme* (e.g. YouTube), which for the acquisition relationship coincide with the two arguments. They do not contribute any bias, since they are replaced with the generic tags $\langle e_1 \rangle$ and $\langle e_2 \rangle$ in all sentences from the bag. There are however other frame elements – *peripheral*, or *extra-thematic* – that can be instantiated with the same value in many sentences. In Figure 1, for instance, sentence S_3 contains two non-core frame elements: the *Means* element (e.g "in a stock-for-stock transaction") and the *Time* element (e.g. "on October 9, 2006"). Words from these elements, like "stock", or "October", are likely to occur very often in the Google-YouTube bag, and because the training dataset contains only a few other bags, subsequence patterns containing these words will be given too much weight in the learned model. This is problematic, since these words can appear in many other frames, and thus the learned model is likely to make errors. Instead, we would like the model to focus on words that trigger the target relationship (in FrameNet, these are the lexical units associated with the target frame).

5.1 A Solution for Type I Bias

In order to account for how strongly the words in a sequence are correlated with either of the individual arguments of the relation, we modify the formula for the sequence weight $\tau(s)$ by factoring in a weight $\tau(w)$ for each word in the sequence, as illustrated in Equation 1.

$$\tau(s) = \lambda^{g(\bar{s})} \cdot \prod_{w \in s} \tau(w) \qquad (1)$$

Given a predefined set of weights $\tau(w)$, it is straightforward to update the recursive computation of the subsequence kernel so that it reflects the new weighting scheme.

If all the word weights are set to 1, then the new kernel is equivalent to the old one. What we want, however, is a set of weights where words that are correlated with either of the two arguments are given lower weights. For any word, the decrease in weight should reflect the degree of correlation between that word and the two arguments. Before showing the formula used for computing the word weights, we first introduce some notation:

- Let $X \in \mathcal{X}$ be an arbitrary bag, and let $X.a_1$ and $X.a_2$ be the two arguments associated with the bag.
- Let $C(X)$ be the size of the bag (i.e. the number of sentences in the bag), and $C(X, w)$ the number of sentences in the bag X that contain the word w. Let $P(w|X) = C(X, w)/C(X)$.
- Let $P(w|X.a_1 \vee X.a_2)$ be the probability that the word w appears in a sentence due only to the presence of $X.a_1$ or $X.a_2$, assuming $X.a_1$ and $X.a_2$ are independent causes for w.

The word weights are computed as follows:

$$\begin{aligned} \tau(w) &= \frac{C(X,w) - P(w|X.a_1 \vee X.a_2) \cdot C(X)}{C(X,w)} \\ &= 1 - \frac{P(w|X.a_1 \vee X.a_2)}{P(w|X)} \end{aligned} \qquad (2)$$

The quantity $P(w|X.a_1 \vee X.a_2) \cdot C(X)$ represents the expected number of sentences in which w would occur, if the only causes were $X.a_1$ or $X.a_2$, independent of each other. We want to discard this quantity from the total number of occurrences $C(X, w)$, so that the effect of correlations with $X.a_1$ or $X.a_2$ is eliminated.

We still need to compute $P(w|X.a_1 \vee X.a_2)$. Because in the definition of $P(w|X.a_1 \vee X.a_2)$, the arguments $X.a_1$ and $X.a_2$ were considered independent causes, $P(w|X.a_1 \vee X.a_2)$ can be computed with the *noisy-or* operator (Pearl, 1986):

$$\begin{aligned} P(\cdot) &= 1-(1-P(w|a_1)) \cdot (1-P(w|a_2)) \qquad (3) \\ &= P(w|a_1)+P(w|a_2)-P(w|a_1) \cdot P(w|a_2) \end{aligned}$$

The quantity $P(w|a)$ represents the probability that the word w appears in a sentence due only to the presence of a, and it could be estimated using counts on a sufficiently large corpus. For our experimental evaluation, we used the following approximation: given an argument a, a set of sentences containing a are extracted from web documents (details in Section 6). Then $P(w|a)$ is simply approximated with the ratio of the number of sentences containing w over the total number of sentences, i.e. $P(w|a) = C(w, a)/C(a)$. Because this may be an overestimate (w may appear in a sentence containing a due to causes other than a), and also because of data sparsity, the quantity $\tau(w)$ may sometimes result in a negative value – in these cases it is set to 0, which is equivalent to ignoring the word w in all subsequence patterns.

6 MIL Relation Extraction Datasets

For the purpose of evaluation, we created two datasets: one for corporate acquisitions, as shown in Table 2, and one for the person-birthplace relation, with the example pairs from Table 3. In both tables, the top part shows the training pairs, while the bottom part shows the test pairs.

$+/-$	**Arg** a_1	**Arg** a_2	**Size**
$+$	Google	YouTube	1375
$+$	Adobe Systems	Macromedia	622
$+$	Viacom	DreamWorks	323
$+$	Novartis	Eon Labs	311
$-$	Yahoo	Microsoft	163
$-$	Pfizer	Teva	247
$+$	Pfizer	Rinat Neuroscience	50 (41)
$+$	Yahoo	Inktomi	433 (115)
$-$	Google	Apple	281
$-$	Viacom	NBC	231

Table 2: Corporate Acquisition Pairs.

$+/-$	**Arg** a_1	**Arg** a_2	**Size**
$+$	Franz Kafka	Prague	552
$+$	Andre Agassi	Las Vegas	386
$+$	Charlie Chaplin	London	292
$+$	George Gershwin	New York	260
$-$	Luc Besson	New York	74
$-$	Wolfgang A. Mozart	Vienna	288
$+$	Luc Besson	Paris	126 (6)
$+$	Marie Antoinette	Vienna	105 (39)
$-$	Charlie Chaplin	Hollywood	266
$-$	George Gershwin	London	104

Table 3: Person-Birthplace Pairs.

Given a pair of arguments (a_1, a_2), the corresponding bag of sentences is created as follows:

■ A query string "$a_1 * * * * * * * a_2$" containing seven wildcard symbols between the two arguments is submitted to Google. The preferences are set to search only for pages written in English, with Safesearch turned on. This type of query will match documents where an occurrence of a_1 is separated from an occurrence of a_2 by at most seven content words. This is an approximation of our actual information need: "return all documents containing a_1 and a_2 in the same sentence".

■ The returned documents (limited by Google to the first 1000) are downloaded, and then the text is extracted using the HTML parser from the Java Swing package. Whenever possible, the appropriate HTML tags (e.g. *BR, DD, P*, etc.) are used as hard end-of-sentence indicators. The text is further segmented into sentences with the OpenNLP[1] package.

■ Sentences that do not contain both arguments a_1 and a_2 are discarded. For every remaining sentence, we find the occurrences of a_1 and a_2 that are closest to each other, and create a relation example by replacing a_1 with $\langle e_1 \rangle$ and a_2 with $\langle e_2 \rangle$. All other occurrences of a_1 and a_2 are replaced with a null token ignored by the subsequence kernel.

The number of sentences in every bag is shown in the last column of Tables 2 & 3. Because Google also counts pages that are deemed too similar in the first 1000, some of the bags can be relatively small.

As described in Section 5.1, the word-argument correlations are modeled through the quantity $P(w|a) = C(w,a)/C(a)$, estimated as the ratio between the number of sentences containing w and a, and the number of sentences containing a. These counts are computed over a bag of sentences containing a, which is created by querying Google for the argument a, and then by processing the results as described above.

7 Experimental Evaluation

Each dataset is split into two sets of bags: one for training and one for testing. The test dataset was purposefully made difficult by including negative bags with arguments that during training were used in positive bags, and vice-versa. In order to evaluate the relation extraction performance at the sentence level, we manually annotated all instances from the positive test bags. The last column in Tables 2 & 3 shows, between parentheses, how many instances from the positive test bags are real positive instances. The corporate acquisition test set has a total of 995 instances, out of which 156 are positive. The person-birthplace test set has a total of 601 instances, and only 45 of them are positive. Extrapolating from the test set distribution, the positive bags in the person-birthplace dataset are significantly sparser in real positive instances than the positive bags in the corporate acquisition dataset.

The subsequence kernel described in Section 4 was used as a custom kernel for the LibSVM[2] Java package. When run with the default parameters, the results were extremely poor – too much weight was given to the slack term in the objective function. Minimizing the regularization term is essential in order to capture subsequence patterns shared among positive bags. Therefore LibSVM was modified to solve the optimization problem from Figure 2, where the capacity parameter C is normalized by the size of the transformed dataset. In this new formulation, C is set to its default value of 1.0 – changing it to other values did not result in significant improvement. The trade-off between false positive and false negative errors is controlled by the parameter c_p. When set to its default value of 0.5, false-negative errors and false positive errors have the same impact on the objective function. As expected, setting c_p to a smaller value (0.1) resulted in better performance. Tests with even lower values did not improve the results.

We compare the following four systems:

■ **SSK–MIL**: This corresponds to the MIL formulation from Section 3, with the original subsequence kernel described in Section 4.

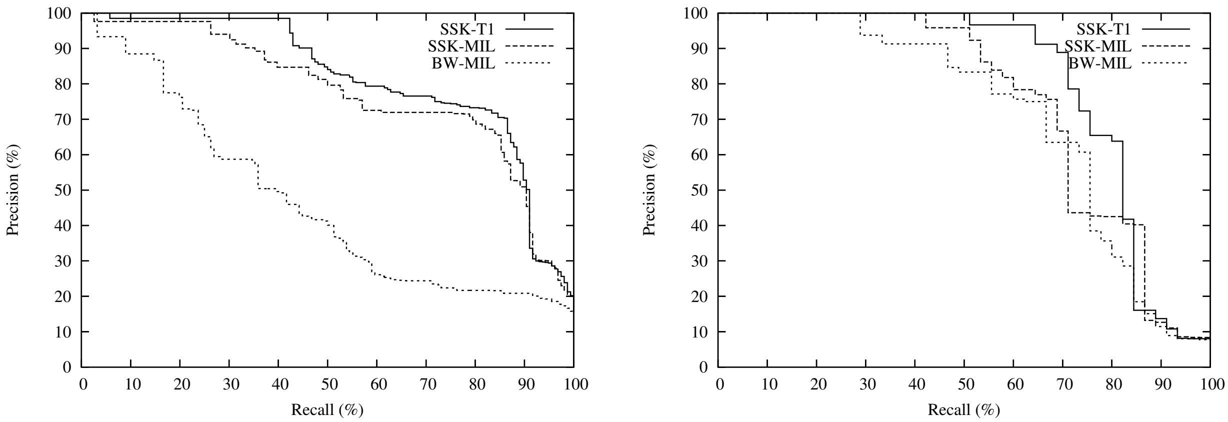
■ **SSK–T1**: This is the SSK–MIL system augmented with word weights, so that the Type I bias is reduced, as described in Section 5.1.

■ **BW-MIL**: This is a bag-of-words kernel, in which the relation examples are classified based on the unordered words contained in the sentence. This baseline shows the performance of a standard text-classification approach to the problem using a state-of-the art algorithm (SVM).

■ **SSK–SIL**: This corresponds to the original subsequence kernel trained with traditional, single instance learning (SIL) supervision. For evaluation, we train on the manually labeled instances from the test bags. We use a combination of one positive bag and one negative bag for training, while the other two bags are used for testing. The results are averaged over all four possible combinations. Note that the supervision provided to SSK–SIL requires sig-

[1] http://opennlp.sourceforge.net

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm



(a) Corporate Acquisitions (b) Person-Birthplace

Figure 3: Precision-Recall graphs on the two datasets.

nificantly more annotation effort, therefore, given a sufficient amount of training examples, we expect this system to perform at least as well as its MIL counterpart.

In Figure 3, precision is plotted against recall by varying a threshold on the value of the SVM decision function. To avoid clutter, we show only the graphs for the first three systems. In Table 4 we show the area under the precision recall curves of all four systems. Overall, the learned relation extractors are able to identify the relationship in novel sentences quite accurately and significantly out-perform a bag-of-words baseline. The new version of the subsequence kernel SSK–T1 is significantly more accurate in the MIL setting than the original subsequence kernel SSK–MIL, and is also competitive with SSK–SIL, which was trained using a reasonable amount of manually labeled sentence examples.

Dataset	SSK–MIL	SSK–T1	BW–MIL	SSK–SIL
(a) CA	76.9%	81.1%	45.9%	80.4%
(b) PB	72.5%	78.2%	69.2%	73.4%

Table 4: Area Under Precision-Recall Curve.

8 Future Work

An interesting potential application of our approach is a web relation-extraction system similar to Google Sets, in which the user provides only a handful of pairs of entities known to exhibit or not to exhibit a particular relation, and the system is used to find other pairs of entities exhibiting the same relation. Ideally, the user would only need to provide positive pairs. Sentences containing one of the relation arguments could be extracted from the web, and likely negative sentence examples automatically created by pairing this entity with other named entities mentioned in the sentence. In this scenario, the training set can contain both false positive and false negative noise. One useful side effect is that Type I bias is partially removed – some bias still remains due to combinations of at least two words, each correlated with a different argument of the relation.

We are also investigating methods for reducing Type II bias, either by modifying the word weights, or by integrating an appropriate measure of word distribution across positive bags directly in the objective function for the MIL problem. Alternatively, implicit negative evidence can be extracted from sentences in positive bags by exploiting the fact that, besides the two relation arguments, a sentence from a positive bag may contain other entity mentions. Any pair of entities different from the relation pair is very likely to be a negative example for that relation. This is similar to the concept of negative *neighborhoods* introduced by Smith and Eisner (2005), and has the potential of eliminating both Type I and Type II bias.

9 Related Work

One of the earliest IE methods designed to work with a reduced amount of supervision is that of Hearst (1992), where a small set of seed patterns is used in a bootstrapping fashion to mine pairs of hypernym-hyponym nouns. Bootstrapping is actually orthogonal to our method, which could be used as the pattern learner in every bootstrapping iteration. A more recent IE system that works by bootstrapping relation extraction patterns from the web is KNOWITALL (Etzioni et al., 2005). For a given target relation, supervision in KNOWITALL is provided as a *rule template* containing words that describe the class of the arguments (e.g. "company"), and a small set of seed extraction patterns (e.g. "has acquired"). In our approach, the type of supervision is different – we ask only for pairs of entities known to exhibit the target relation or not. Also, KNOWITALL requires large numbers of search engine queries in order to collect and validate extraction patterns, therefore experiments can take weeks to complete. Comparatively, the approach presented in this paper requires only a small number of queries: one query per relation pair, and one query for each relation argument.

Craven and Kumlien (1999) create a noisy training set for the subcellular-localization relation by mining Medline for sentences that contain tuples extracted from relevant medical databases. To our knowledge, this is the first approach that is using a "weakly" labeled dataset for relation extraction. The resulting bags however are very dense in positive examples, and they are also many and small – consequently, the two types of bias are not likely to have significant impact on their system's performance.

10 Conclusion

We have presented a new approach to relation extraction that leverages the vast amount of information available on the web. The new RE system is trained using only a handful of entity pairs known to exhibit and not exhibit the target relationship. We have extended an existing relation extraction kernel to learn in this setting and to resolve problems caused by the minimal supervision provided. Experimental results demonstrate that the new approach can reliably extract relations from web documents.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful suggestions. This work was supported by grant IIS-0325116 from the NSF, and a gift from Google Inc.

References

Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Support vector machines for multiple-instance learning. In *NIPS 15*, pages 561–568, Vancouver, BC. MIT Press.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proc. of COLING–ACL '98*, pages 86–90, San Francisco, CA. Morgan Kaufmann Publishers.

Razvan C. Bunescu and Raymond J. Mooney. 2006. Subsequence kernels for relation extraction. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *NIPS 18*.

M. Craven and J. Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proc. of ISMB'99*, pages 77–86, Heidelberg, Germany.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proc. of ACL'04*, pages 423–429, Barcelona, Spain, July.

Thomas G. Dietterich, Richard H. Lathrop, and Tomas Lozano-Perez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence*, 165(1):91–134.

T. Gartner, P.A. Flach, A. Kowalczyk, and A.J. Smola. 2002. Multi-instance kernels. In *In Proc. of ICML'02*, pages 179–186, Sydney, Australia, July. Morgan Kaufmann.

M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of ACL'92*, Nantes, France.

Judea Pearl. 1986. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288.

Soumya Ray and Mark Craven. 2005. Supervised versus multiple instance learning: An empirical comparison. In *Proc. of ICML'05*, pages 697–704, Bonn, Germany.

Bernhard Schölkopf and Alexander J. Smola. 2002. *Learning with kernels - support vector machines, regularization, optimization and beyond*. MIT Press, Cambridge, MA.

N. A. Smith and J. Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proc. of ACL'05*, pages 354–362, Ann Arbor, Michigan.

Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. John Wiley & Sons.

D. Zelenko, C. Aone, and A. Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

Q. Zhang, S. A. Goldman, W. Yu, and J. Fritts. 2002. Content-based image retrieval using multiple-instance learning. In *Proc. of ICML'02*, pages 682–689.