Mining the Web for Relations between Digital Devices using a Probabilistic Maximum Margin Model

Oksana Yakhnenko
Iowa State University
Ames, IA, 50010
oksayakh@cs.iastate.edu

Barbara Rosario
Intel Research
Santa Clara, CA, 95054
barbara.rosario@intel.com

Abstract

Searching and reading the Web is one of the principal methods used to seek out information to resolve problems about technology in general and digital devices in particular. This paper addresses the problem of text mining in the digital devices domain. In particular, we address the task of detecting semantic relations between digital devices in the text of Web pages. We use a Naïve Bayes model trained to maximize the margin and compare its performance with several other comparable methods. We construct a novel dataset which consists of segments of text extracted from the Web, where each segment contains pairs of devices. We also propose a novel, inexpensive and very effective way of getting people to label text data using a Web service, the Mechanical Turk. Our results show that the maximum margin model consistently outperforms the other methods.

1 Introduction

In the digital home domain, home networks are moving beyond the common infrastructure of routers and wireless access points to include application-oriented devices like network attached storage, Internet telephones (VOIP), digital video recorders (e.g., Tivo), media players, entertainment PCs, home automation, and networked photo printers. There is an ongoing challenge associated with domestic network design, technology education, device setup, repair, and tuning. In this digital home setting, searching the Web is one of the principle methods used to seek out information and to resolve problems about technology in general and about digital devices in particular (Bly et al., 2006).

This paper addresses the problem of automatic text mining in the digital networks domain. Understanding the relations between entities in natural language sentences is a crucial step toward the goal of text mining. We address the task of identifying and extracting the sentences from Web pages which expressed a relation between two given digital devices in contrast to sentences in which these devices co-occur.

As an example, consider a user who is looking for information on digital video recorders (DVR), in particular, on how she can use a DVR with a PC. This user will not be satisfied with finding Web pages that simply mention these devices (such as the many products catalogs or shopping sites), but rather, the user is interested in retrieving and reading only the Web pages in which a specific relation between the two devices is expressed. The user is interested to learn that, for example, *"Any modern Windows PC can be used for DVR duty"* or that it is possible to transfer data from a DVR to a PC (*"You can simply take out the HD from the DVR, hook it up to the PC, and copy the videos over to the PC"*).[1]

The specific task addressed in this paper is the following: given a pair of devices, search the Web and extract only the sentences in which the devices are actually involved in an activity or a relation in the retrieved Web pages.

Note that we do not attempt to identify the type

---

[1] In italic are real sentences extracted from Web pages.

of relationship between devices but rather we classify sentences into whether the relation or activity is present or not, and thus we frame the problem as a binary text classification problem.[2] We propose a directed maximum margin probabilistic model to solve this classification task. Maximum margin probabilistic models have received a lot of attention in the machine learning and natural language processing literature. These models are trained to maximize the smallest difference between the probabilities of the true class and the best alternative class. Approaches such as maximum margin Markov networks (M3N) (Taskar et al., 2003) have been considered in prediction problems in which the goal is to assign a label to each word in the sentence or a document (such as part of speech tagging). It has also been shown that training of Bayesian networks by maximizing the margin can result in better performance than M3N in a flat-table structured domain (simulated and UCI repository datasets) and a structured prediction problem (protein secondary structure) (Guo et al., 2005). Given this background, we draw our attention to the application of maximum margin probabilistic models to a text classification task. We consider a *directed* model, where the parameters represent a probability distribution for words in each class (maximum margin equivalent of a Naïve Bayes). We evaluate the maximum margin model and compare its performance with the equivalent joint likelihood model (Naïve Bayes), conditional likelihood model (logistic regression) and support vector machines (SVM) on the relationship extraction task described above, as well as several other classification methods. Our results show that the maximum margin Naïve Bayes outperforms the other methods in terms of classification accuracy. To train such a model, manually labeled data is required, which is usually slow and expensive to acquire. To address this, we propose a novel, inexpensive and very effective way of getting people to label text data using the Mechanical Turk, an Amazon website[3] where people earn "micro-money" for

completing tasks which are simple for humans to accomplish.

The paper is organized as follows: in Section 2 we discuss related work. In Section 3 we review joint likelihood and conditional likelihood models and maximum margin Naïve Bayes. In Section 4 we describe the collection of the training sentences, and how Mechanical Turk was used to construct the labels for the data. Section 5 introduces the experimental setup and presents performance results for each of the algorithms. We analyze Naïve Bayes, maximum margin Naïve Bayes and logistic regression in terms of the learned probability distributions in Section 6. Section 7 concludes with discussion.

2 Related work

2.1 Relation extraction

There has been a spate of work on relation extraction in recent years. However, many papers actually address the task of role extraction: (usually two) entities are identified and the relationship is *implied* by the co-occurrence of these entities or by some linguistic expression (Agichtein and Gravano, 2000; Zelenko et al., 2003).

Several papers propose the use of machine learning models and probabilistic models for relation extraction: Naïve Bayes for the relation *subcellular-location* in the bio-medical domain (Craven, 1999) or for *person-affiliation* and *organization-location* (Zelenko et al., 2003). Rosario and Hearst (2005) have used a more complicated dynamic graphical model to identify interaction types between proteins and to simultaneously extract the proteins.

2.2 Maximum margin models

Probabilistic graphical models and different approaches to training them have received a lot of attention in application to natural language processing. McCallum and Nigam (1998) showed that Naïve Bayes can be a very accurate model for text categorization.

Since probabilistic graphical models represent joint probability distributions whereas classification focuses on the conditional probability, there has been debate regarding the objective that should be maximized in order to train these models. Ng and Jordan (2001) have compared a joint likelihood

---

[2] Classifying or clustering the relation types would involve the tricky task of defining the possible semantic relations between devices as well as relations. We plan of addressing this in the future work, however, we believe that such binary distinction is already quite useful for many tasks in this domain.

[3] Available at http://www.mturk.com

model (Naïve Bayes) and its discriminative counterpart (logistic regression), and they have shown that while for large number of examples logistic regression has a lower error rate, Naïve Bayes often outperforms logistic regression for smaller data sets. However, Klein and Manning (2002) showed that for natural language and text processing tasks, conditional models are usually better than joint likelihood models. Yakhnenko et al. (2005) also showed that conditional models suffer from overfitting in text and sequence structured domains.

In recent years, the interest in learning parameters of probabilistic models by maximizing the probabilistic margin has developed. Taskar et al. (2003) have solved the problem of learning Markov networks (undirected graphs) by maximizing the margin. Their work has focused on likelihood based structured classification where the goal is to assign a class to each word in the sentence or a document. Guo et al. (2005) have proposed a solution to learning parameters of the maximum margin Bayesian Networks.

Surprisingly, little has been done in applying probabilistic models trained to maximize the margin to simple classification tasks (to the best of our knowledge). Therefore, since the Naïve Bayes model has been shown to be a successful algorithm for many text classification tasks (McCallum and Nigam, 1998) we suggest learning the parameters of Naïve Bayes model to maximize the probabilistic margin. We apply the Naïve Bayes model trained to maximize the margin to a relation extraction task.

3 Joint and conditional likelihood models and maximum margin

We now describe the background in probabilistic models as well as different approaches to parameter estimation for probabilistic models. In particular, we describe Naïve Bayes, logistic regression (analogous to conditionally trained Naïve Bayes) and then introduce Naïve Bayes trained to maximize the margin.

First, we introduce some notation. Let D be a corpus that consists of training examples. Let T be the size of D. We represent each example with a tuple $\langle s, c \rangle$ where s is a sentence or a document, and c is a label from a set of all possible labels, $c \in C =$ $\{c_1 ... c_m\}$. Let $D = \left\{ \left\langle s^i, c^i \right\rangle \right\}$ where superscript $1 \leq i \leq T$ is the index of the document in the corpus, and c^i is the label of example s^i. Let V be vocabulary of D, so that every document s consists of elements of V. We will use s_j to denote a word from s in position j, where $1 \leq j \leq length(s)$.

3.1 Generative and discriminative Naïve Bayes models

A probabilistic model assigns to each instance s a joint probability of the instance and the class $P(s, c)$. If the probability distribution is known, then a new instance s_{new} can be classified by giving it a label which has the highest probability:

$$c = \arg \max_{c_k \in C} P(c_k | s_{new}) \qquad (1)$$

Joint likelihood models learn the parameters by maximizing the probability of an example and its class, $P(s, c)$. Naïve Bayes multinomial, for instance, assumes that all words in the sentence are independent given the class, and computes this probability as $P(c) \prod_{j=1}^{length(s)} P(s_j | c)$. Each of $P(s_j | c)$ and $P(c)$ are estimated from the training data using relative frequency estimates. From here on we will refer to joint likelihood Naïve Bayes multinomial as NB-JL.

Since the conditional probability is needed for the classification task, it has been suggested to solve the maximization problem and train the model so that the choice of the parameters maximizes $P(c|s)$ directly. One can use a joint likelihood model to obtain joint probability distribution $P(s, c)$ and then use the definition of conditional probability to get $P(c|s) = P(s, c) / \sum_{c_k \in C} P(s, c_k)$. The solutions that maximize this objective function are searched for by using gradient ascent methods. Logistic regression is a conditional model that assumes the independence of features given the class, and it is a conditional counterpart to NB-JL (Ng and Jordan, 2001).

We will now introduce a probabilistic maximum margin objective and describe a maximum margin model that is analogous to Naïve Bayes and logistic regression.

3.2 Maximum margin training of Naïve Bayes models

The basic idea behind maximum margin models is to choose model parameters that for each example will make the probability of the true class and the example as high as possible while making the probability of the nearest alternative class as low as possible. Formally, the maximum margin objective is

$$\gamma = \min_{i=1}^{T} \min_{c \neq c^i} \frac{P(c^i|s^i)}{P(c|s^i)} = \min_{i=1}^{T} \min_{c \neq c^i} \frac{P(s^i, c^i)}{P(s^i, c)} \quad (2)$$

Here $P(s, c)$ is modeled by a generative model, and parameter learning is reduced to solving a convex optimization problem (Guo et al., 2005).

In order for the example to be classified correctly, the probability of the true class given the example has to be higher than the probability of getting the wrong class or

$$\gamma_i = \log p(c^i|s^i) - \log p(c^j|s^i) > 0 \quad (3)$$

where $j \neq i$ and c^i is the true label of example s^i. The larger the margin γ_i is, the more confidence we have in the prediction.

We consider a Naïve Bayes model trained to maximize the margin and refer to this model as MMNB. Using exponential family notation, let $P(s_j|c) = e^{\mathbf{w}_{s_j|c}}$. The likelihood is $P(s, c) = e^{\mathbf{w}_c} \prod_{j=1}^{len(s)} e^{\mathbf{w}_{s_j|c}}$. Then the log-likelihood

$$\log P(s, c) = \mathbf{w_c} + \sum_{j=1}^{len(s)} count(s_j)\mathbf{w}_{s_j|c} = \mathbf{w} \cdot \phi(s, c) \quad (4)$$

where $\mathbf{w}$ is the weight vector for all the parameters that need to be learned, and $\phi(s, c)$ is the vector of counts of words associated with each parameter $\phi(s, c) = (...count(s_j c)....)$ in s for class c.

The general formulation for Bayesian networks was given in Guo et al., and we adapt their formulation for training a Naïve Bayes model. The parameters are learned by solving a convex optimization problem. If the margin γ is the smallest log-ratio, then γ needs to be maximized, where the constraint is that for each instance the log-ratio of the probability of predicting the instance correctly and predicting it incorrectly is at least γ. Such formulation also allows for the use of slack variables ξ so that the classifier "gives up" on the examples that are difficult to classify.

$$\text{minimize }_{\gamma, \mathbf{w}, \xi} \frac{1}{\gamma^2} + B \sum_{i=1}^{T} \xi_i$$

$$\text{subject to } \mathbf{w}(\phi(i, c^i) - \phi(i, c)) \geq \gamma \delta(c^i, c) - \xi_i$$

$$\text{and } \sum_{s_i \in V} e^{\mathbf{w}_{s_i, c}} \leq 1 \forall c \in C$$

$$\text{and } \gamma \geq 0$$

This problem is convex in the variables $\gamma, \mathbf{w}, \epsilon$. B is a regularization parameter, and $\delta(c^i, c) = 1$ if $c^i \neq c$ and 0 otherwise. The inequality constraint for probabilities is needed to preserve convexity of the problem, and in the case of Naïve Bayes, the probability distribution over the parameters (the equality constraint) can be easily obtained by renormalizing the learned parameters.

The minimization problem is somewhat similar to ℓ_2-norm support vector machine with a soft margin (Cristianini and Shawe-Taylor, 2000). The first constraint imposes that for each example the log of the ratio between the example under the true class and the example under some alternative class is greater than the margin allowing for some slack. The second constraint enforces that the parameters do not get very large and that the probabilities sum to less than 1 to maintain valid probability distribution (the inequality constraint is required to preserve convexity, and the probability distribution can be obtained after training by renormalization).

Following Guo et al. (2005), we find parameters using a log-barrier method (Boyd and Vandenberghe, 2004), the sum of the logarithms of constraints are subtracted from the objective and scaled by a parameter μ. The problem is solved sequentially using a fixed μ and gradually lowering μ to 0. The solution for a fixed μ is obtained using (typically) a second order method to guarantee faster convergence. This solution is then used as the initial parameter values for the next μ. In our implementation we used a limited memory quasi-Newton method (Nocedal and Liu, 1989).

4 Data and labels

4.1 The problem of labeling data

One major problem of natural language processing is the sparsity of data; to accurately learn a linguistic model, one needs to label a large amount of text, which is usually an expensive requirement. For information extraction, the labeling process is particularly difficult and time consuming. Moreover, in different applications one needs different labeled data for each domain. We propose a creative way of convincing many people to label data quickly and at low cost to us by using the Mechanical Turk. Similarly, Luis von Ahn (2006) creates very successful and compelling computer games in such a way that while playing, people provide labels for images on the Web.

4.2 Collecting data and label agreement analysis

To collect the data, we identified 58 pairs of digital devices, as well as their synonyms (for example, computer, laptop, PC, desktop, etc), and different manufacturers for a given device (for example Toshiba, Dell, IBM, etc). The devices alone were used to construct the query (for example 'computer, camera', as well as a combination of manufacturer and devices (for example 'dell laptop, cannon camera'). Each of these pairs was used as a query in Google, and the sentences that contain both devices were extracted resulting in a total of 3624 sentences. We use the word 'sentence' when referring to the examples, however we note that not all text excerpts are sentences, some are chunks of text data.

To label the data we used the Mechanical Turk (MTurk), a Web service that allows you to create and post a task for humans to solve; typical tasks are labeling pictures, choosing the best among several photographs, writing product descriptions, proofreading and transcribing podcasts. After the task is completed the requesters can then review the submissions and reject them if the results are poor.

We created a total of 121 unique surveys consisting of 30 questions. Each question consisted of one of the extracted statements with the devices highlighted in red. The task for the labeler was to choose between 'Yes', if the statement contained a relation between the devices, 'No' if it did not, or 'not ap-

| | | worker3 | | |
worker1	worker2	yes	no	n/a
yes	yes	**1091**	237	23
	no	226	281	22
	n/a	19	18	6
no	yes	217	199	8
	no	186	**870**	56
	n/a	14	39	8
n/a	yes	17	13	5
	no	6	32	6
	n/a	4	12	**9**

Table 1: Summary of the labels assigned by the MT workers to all the sentences.

plicable' if the text extract was not a sentence, or if the query words were not used as different devices (as for noun compounds such as *computer stereo*).[4] Each survey was assigned to 3 distinct workers, thus having 3 possible labels for all 3624 sentences.[5]

We used Fleiss's kappa (Fleiss, 1971) (a generalization of kappa statistic which takes into account multiple raters and measures inter-rater reliability) in order to determine the degree of agreement and to determine whether the agreement was accidental. Kappa statistics is a number between 0 and 1 where 0 is random agreement, and 1 is perfect agreement.

In order to compute kappa statistic, since the computation requires that the raters are the same for each survey, we mapped workers into 'worker1', 'worker2', 'worker3' with 'worker1' being the first worker to complete each of the 121 surveys, 'worker2' the second, and so on. The responses are summarized in Table 1.

The overall Fleiss's kappa was 0.41[6], and therefore, it can be concluded that the agreement between the workers was not accidental.

We had perfect agreement for 49% of all sentences, 5% received all three labels (these examples were discarded) and for the remaining 46% two la-

---

[4]This dataset, including all the MTurk's workers responses is available at http://www.cs.iastate.edu/~oksayakh/relation_data.html

[5]The requirement for the workers to be different was imposed by the MTurk system, which checks their Amazon identity; however, this still allows for the same person who has multiple identities to complete the same task more than once.

[6]The kappa coefficients for categories 'Yes' and 'No' were 0.45 and 0.41 respectively (moderate agreement) and for category 'not applicable' was 0.15 (slight agreement).

bels were assigned (the majority vote was used to determine the final label). For these cases, we noticed that some of the labels were wrong (however in most cases the majority vote results in the correct label) but other sentences were ambiguous and either label could be right. To assign the final label we used majority vote, and we discarded sentences for which 'not applicable' was the majority label.

We rewarded the users with between 15 and 30 cents per survey (resulting in less than a cent for a text segment) and we were able to obtain labels for 3594 text segments for under $70. It also took anywhere between a few minutes to a half-hour from the time the survey was made available until it was completed by all three users. We find Mechanical Turk to be a quite interesting, inexpensive, fairly accurate and fast way to obtain labeled data for natural language processing tasks.

We used this data to evaluate the classification models as described in the next section.

5   Experimental setup and results

The words were stemmed, and the data was smoothed by mapping all the words that appeared only once to a unique token `smoothing_token` (resulting in a total of approximately 2,800 words in the vocabulary). We performed 10-fold cross-validation, with smoothed test data where all the unseen words in the test data were mapped to the token `smoothing_token`. We used the exact same data in the folds for all four algorithms – MMNB, NB-JL, logistic regression and SVM. Since MMNB, SVM, and logistic regression allows for regularization, we used tuning to find the optimal performance of the models. At each fold we withheld 30% of the training data for validation purposes (thus resulting in 3 disjoint sets at each fold). The model was trained on the resulting 70% of the training data for different values of the regularization parameters, and the value which yielded the highest accuracy on the validation set was used to train the model that was evaluated on the test set.

As a baseline, we consider a classifier which assigns the most frequent label ('Yes'); such a classifier results in 53% accuracy.

Table 2 summarizes the performance of MMNB and other algorithms as determined by 10-fold cross-

Algorithm	Accuracy
MMNB	**80.23%**
SVM-RBF	76.49%
NB-JL	75.62%
Perceptron	74.04%
SVM-2	72.72%
SVM-3	71.54%
DT	70.76%
LR	69.95%
SVM-1	69.94%
Baseline	53.8%

Table 2:   Classification accuracies as determined by 10-fold cross-validation. SVM-1 uses linear kernel, SVM-2 uses quadratic kernel, SVM-3 uses cubic kernel, SVM-RBF uses RBF kernel with parameter $\gamma = 0.1$. The Decision Tree (DT) uses binary splits. LR is logistic regression.

validation with tuning data. We compared the accuracies of the maximum margin model with the accuracy of generative Naïve Bayes, logistic regression and SVM as shown in Table 2. The MMNB has the highest accuracy followed by NB-JL and then SVM with RBF kernel. Even after tuning, logistic regression did not reach the performance of MMNB and NB-JL.

Since MMNB is trained to maximize the margin, we compared it with the Support Vector Machine (linear maximum margin classifier). Counts of words were used as features (resulting in the bag of words representation[7]). We ran our experiments with linear, quadratic, cubic and RBF kernels. SVM was tuned using the validation set similarly to MMNB. We also experimented with Perceptron and Decision Tree using binary splits with reduced error-pruning, which are methods commonly used for text classification (due to lack of space, we will not describe these methods and their applications, but refer the reader to Manning and Schütze (1999)). Among all the known methods, the maximum margin Naïve Bayes is the algorithm with the highest accuracy, suggesting that it is a competitive algorithm in relation extraction and text classification tasks.

---

[7]This representation allows for additional or alternative features such as k-grams of words, whether the words are capitalized, where on the page the sentence was located, etc. Evaluating MMNB and other methods with additional features is of interest in the future

6 Analysis of behavior of Naïve Bayes, maximum margin Naïve Bayes and logistic regression

We analyzed the behavior of the parameters of the probabilistic models (Naïve Bayes, MMNB and logistic regression) on the training data. For each example in the training data we computed the probability $P(c = noRelation|s)$ using the parameters from the model, and examined the probabilities assigned to examples from both classes. We show these plots in Figure 1.
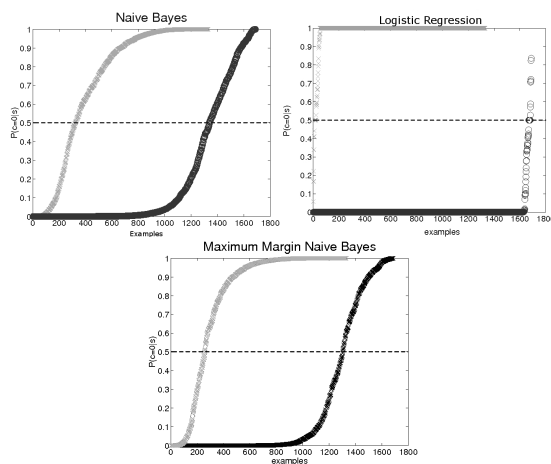

Figure 1: Probability distribution of $P(c = noRelation|s)$ learned by the Naïve Bayes (upper left), logistic regression (upper right) and maximum margin Naïve Bayes(lower). In gray are class-conditional probabilities assigned to positive examples, and in black are class-conditional probabilities assigned to negative examples.

As we see, the logistic regression discriminates between the majority of the examples by assigning extreme probabilities (0 and 1). However, there are some examples which are extremely borderline, and thus it does not generalize well on the test set. On the other had, Naïve Bayes does not have such "sharp" discrimination. Maximum margin Naïve Bayes has "sharper" discrimination than Naïve Bayes, however the discrimination is smoother than for logistic regression. The examples which are more difficult to classify have probabilities that are more spread out (away from 0.5), as opposed to the case of logistic regression, which assigns these difficult examples to probability close to 0.5. This suggests that maximum margin Naïve Bayes, possibly has a better generalization ability than both logistic regression and

Naïve Bayes, however to make such a claim additional experiments are needed.

7 Conclusions

The contribution of this paper is threefold. First, we addressed the important problem of identifying the presence of semantic relations between entities in text, focusing on the digital domain. We presented some encouraging results; it remains to be seen however, how this would transfer to better results in an information retrieval task. Secondly, we considered a probabilistic model trained to maximize the margin, that achieved the highest accuracy for this task, suggesting that it could be a competitive algorithm for relation extraction and text classification in general. However in order to fully evaluate the MMNB method for relation classification it needs to be applied to other classification and or relation prediction tasks. We also empirically analyzed the behavior of the parameters learned by maximum margin model and showed that the parameters allow for better generalization power than Naïve Bayes or logistic regression models. Finally, we suggested an inexpensive way of getting people to label text data via Mechanical Turk.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of Digital Libraries*.

Sara Bly, William Schilit, David McDonald, Barbara Rosario, and Ylian Saint-Hilaire. 2006. Broken expectations in the digital home. In *Proceedings of Computer Human Interaction (CHI)*.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

Mark Craven. 1999. Learning to extract relations from Medline. In *AAAI-99 Workshop*.

Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Yuhong Guo, Dana Wilkinson, and Dale Schuurmans. 2005. Maximum margin bayesian networks. In *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, page 233.

Dan Klein and Christopher Manning. 2002. Conditional structure versus conditional estimation in nlp models. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, June.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*.

Andrew Y. Ng and Michael I. Jordan. 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Proceedings of Neural Information Processing Systems (NIPS)*, pages 841–848.

Jorge Nocedal and Dong C. Liu. 1989. On the limited memory method for large scale optimization. *Mathematical Programming*, 3(45):503–528.

Barbara Rosario and Marti Hearst. 2005. Multi-way relation classification: Application to protein-protein interactions. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Benjamin Taskar, Carlos Guestrin, and Daphne Koller. 2003. Max-margin markov networks. In *Proceedings of Neural Information Processing Systems (NIPS)*.

Luis von Ahn. 2006. Games with a purpose. *Computer*, 39(6):92–94.

Oksana Yakhnenko, Adrian Silvescu, and Vasant Honavar. 2005. Discriminatively trained markov model for sequence classification. In *Proceedings of International Conference on Data Mining (ICDM)*.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.