Probabilistic Reasoning for Entity & Relation Recognition

Dan Roth **Wen-tau Yih**

Department of Computer Science
University of Illinois at Urbana-Champaign
{danr, yih}@uiuc.edu

Abstract

This paper develops a method for recognizing relations and entities in sentences, while taking mutual dependencies among them into account. E.g., the *kill (KFJ, Oswald)* relation in: "`J. V. Oswald was murdered at JFK after his assassin, R. U. KFJ...`" depends on identifying Oswald and KFJ as *people*, JFK being identified as a *location*, and the *kill* relation between Oswald and KFJ; this, in turn, enforces that Oswald and KJF are *people*.

In our framework, classifiers that identify entities and relations among them are first learned from local information in the sentence; this information, along with constraints induced among entity types and relations, is used to perform global inference that accounts for the mutual dependencies among the entities.

Our preliminary experimental results are promising and show that our global inference approach improves over learning relations and entities separately.

1 Introduction

Recognizing and classifying entities and relations in text data is a key task in many NLP problems, such as information extraction (IE) (Califf and Mooney, 1999; Freitag, 2000; Roth and Yih, 2001), question answering (QA) (Voorhees, 2000) and story comprehension (Hirschman et al., 1999). In a typical IE application of constructing a job database, the system has to extract meaningful entities like *title* and *salary*, and it needs to determine whether the entities are associated with the same position. In a QA system, many questions ask for specific entities involved in some relations. For example, the question "Where was Poe born?" in TREC-9 asks for the *location* entity in which Poe was *born*. The question "Who killed Lee Harvey Oswald?" seeks a *person* entity that has the relation *kill* with the *person* Lee Harvey Oswald.

In all cases we know of, the tasks of identifying entities and relations are treated as separate problems. The common procedure is to first identify and classify entities using a named entity recognizer and only then determine the relations between the entities. However, this approach has several problems. First, if the named entity recognizer is not perfect, the errors it makes will propagate to the relation classifier. For example, if "Boston" is mislabelled as a person, it will never be classified as the location of Poe's birthplace. Second, relation information is sometimes crucial to resolving ambiguous named entity recognition. For instance, if the information that entity "JFK" is the victim of the assassination is given, the named entity recognizer is unlikely to misclassify it as a location (e.g. JFK airport).

This paper develops a novel approach for this problem – a probabilistic framework for recognizing entities and relations together. In this framework, separate classifiers are first trained for entities and relations. Their output is used to represent a conditional distribution for each entity and relation, given the observed data. This information, along with constraints induced among relations and entities (e.g. the first argument of *kill* is a *person*; the second argument of *born_in* is a *location*) are used to make global inferences for the most probable assignment for both entities and relations.

The rest of the paper is organized as follows. Section 2 defines the problem in a formal way. Section 3 describes our approach to this problem. It first introduces how we learn the classifiers, and then introduces the belief network we use to reason for global predictions. Section 4 records the preliminary experiments we did and exhibits some promising results. Finally, section 5 discusses some of the open problems and future work in this framework.

2 Global Inference of Entities/Relations

The problem at hand is that of producing a coherent labeling of entities and relations in given a sentence. Conceptually, the entities and relations can be viewed, taking into account the mutual dependencies, as the graph in Figure 1, where the nodes represent entities (e.g. phrases) and the links denote the binary relations between the entities. Each entity and relation has several properties. Some of the properties, such as words inside the entities and pos tags of words in the context of the sentence, are easy to acquire. However, other properties like the semantic types (i.e., class labels, such as "people", "locations") of phrases are difficult. Identifying the labels of entities and relations is treated here as a learning problem. In particular, we learn these target properties as functions of all other properties of the sentence.

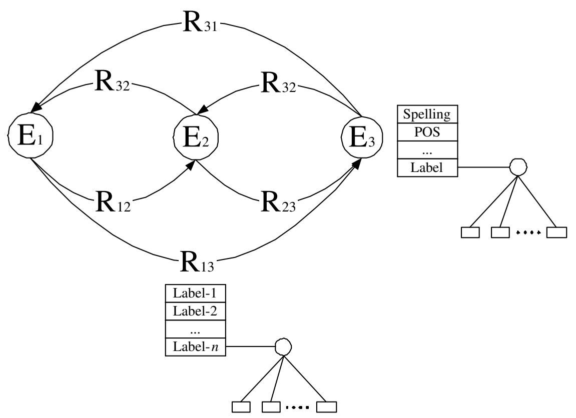

Figure 1: Conceptual view of entities and relations

To describe the problem in a formal way, we first define sentences and entities as follows.

Definition 2.1 (Sentence & Entity) *A sentence S is a linked list which consists of words w and entities E. An entity can be a single word or a set of consecutive words with a predefined boundary. Entities in a sentence are labeled as $E_1, E_2, \cdots$ according to their order, and they take values that range over a set of entity types C^E.*

Notice that determining the entity boundaries is also a difficult problem – the *segmentation* (or *phrase detection*) problem (Abney, 1991; Punyakanok and Roth, 2001). Here we assume it is solved and given to us as input; thus we only concentrate on classification.

Dole 's wife , Elizabeth , is a native of Salisbury , N.C.

E1 **E2** **E3**

Figure 2: A sentence that has three entities

Example 2.1 *The sentence in Figure 2 has three entities: E_1 = "Dole", E_2 = "Elizabeth", and E_3 = "Salisbury, N.C."*

A relation is defined by the entities that are involved in it (its arguments). In this paper, we only discuss binary relations.

Definition 2.2 (Relation) *A (binary) relation $R_{ij} = (E_i, E_j)$ represents the relation between E_i and E_j, where E_i is the first argument and E_j is the second. In addition, R_{ij} can range over a set of entity types C^R.*

Example 2.2 *In the sentence given in Figure 2, there are six relations between the entities: R_{12} = ("Dole", "Elizabeth"), R_{21} = ("Elizabeth", "Dole"), R_{13} = ("Dole", "Salisbury, N.C."), R_{31} = ("Salisbury, N.C.", "Dole"), R_{23} = ("Elizabeth", "Salisbury, N.C."), and R_{32} = ("Salisbury, N.C.", "Elizabeth")*

We define the types (i.e. classes) of relations and entities as follows.

Definition 2.3 (Classes) *We denote the set of predefined entity classes and relation classes as C^E and C^R respectively. C^E has one special element* other_ent*, which represents any unlisted entity class. Similarly, C^R also has one special element* other_rel*, which means the involved entities are irrelevant or the relation class is undefined.*

When clear from the context, we use E_i and R_{ij} to refer to the entity and relation, as well as their types (class labels).

Example 2.3 *Suppose C^E =* { other_ent, person, location } *and C^R =* { other_rel, born_in, spouse_of }. *For the entities in Figure 2, E_1 and E_2 belong to* person *and E_3 belongs to* location. *In addition, relation R_{23} is* born_in, *R_{12} and R_{21} are* spouse_of. *Other relations are* other_rel.

The class label of a single entity or relation depends not only on its local properties, but also on properties of other entities and relations. The classification task is somewhat difficult since the predictions of entity labels and relation labels are mutually dependent. For instance, the class label of E_1 depends on the class label of R_{12} and the class label of R_{12} also depends on the class label of E_1 and E_2. While we can assume that all the data is annotated for training purposes, this cannot be assumed at evaluation time. We may presume that some local properties such as the word, pos, etc. are given, but none of the class labels for entities or relations is.

To simplify the complexity of the interaction within the graph but still preserve the characteristic of mutual dependency, we abstract this classification problem in the following probabilistic framework. First, the classifiers are trained independently and used to estimate the probabilities of assigning different labels given the observation (that is, the easily classified properties in it). Then, the output of the classifiers is used as a conditional distribution for each entity and relation, given the observation. This information, along with the constraints among the relations and entities, is used to make global inferences for the most probable assignment of types to the entities and relations involved.

The class labels of entities and relations in a sentence must satisfy some constraints. For example, if E_1, the first argument of R_{12}, is a *location*, then R_{12} cannot be *born_in* because the first argument of relation *born_in* has to be a *person*. We define constraints as follow.

Definition 2.4 (Constraint) *A constraint $\mathcal{C}$ is a 3-tuple $(\mathcal{R}, \mathcal{E}^1, \mathcal{E}^2)$, where $\mathcal{R} \in C^R$ and $\mathcal{E}^1, \mathcal{E}^2 \in C^E$. If the* *class label of a relation is $\mathcal{R}$, then the legitimate class labels of its two entity arguments are $\mathcal{E}^1$ and $\mathcal{E}^2$ respectively.*

Example 2.4 *Some examples of constraints are: (*born_in, person, location*), (*spouse_of, person, person*), and (*murder, person, person*)*

The constraints described above can be easily written as conditional probabilities. In particular, the probability $P(R_{ij}|E_i, E_j)$ has the following properties.

Property 1 *The probability of the label of relation R_{ij} given the labels of its arguments E_i and E_j has the following properties.*

- $P(R_{ij} = \text{other\_rel}|E_i = e^1, E_j = e^2) = 1$, *if there exists no r, such that (r, e^1, e^2) is a constraint.*
- $P(R_{ij} = r|E_i = e^1, E_j = e^2) = 0$, *if there exists no constraint c, such that $c = (r, e^1, e^2)$.*

Note that the conditional probabilities do not need to be specified manually. In fact, they can be easily learned from an annotated training dataset.

Under this framework, finding the most suitable coherent labels becomes the problem of searching the most probable assignment to all the E and R variables. In other words, the global prediction $e_1, e_2, ..., e_n, r_{12}, r_{21}, ..., r_{n(n-1)}$ satisfies the following equation.

$$(e_1, ..., e_n, r_{12}, ..., r_{n(n-1)}) = \arg\max_{e_i, r_{jk}} Prob(E_1, ..., E_n, R_{12}, ..., R_{n(n-1)}).$$

3 Computational Approach

Each nontrivial property of the entities and relations, such as the class label, depends on a very large number of variables. In order to predict the most suitable coherent labels, we would like to make inferences on several variables. However, when modelling the interaction between the target properties, it is crucial to avoid accounting for dependencies among the huge set of variables on which these properties depend. This is because incorporating these dependencies into our inference is unnecessary and will make the inference intractable. Instead, we can abstract these dependencies in a way by learning the probability of each property conditioned upon an observation. The number of features on which this learning problem depends could be huge, and they can be of different granularity and based on previous learned predicates (e.g. pos), as abstracted into the triangles in Figure 1. Inference is then made based on the probabilities. This approach is similar to (Punyakanok and Roth, 2001; Lafferty et al., 2001) only that there it is restricted to sequential inference, and done only for syntactic structures.

The following subsections describe the details of these two stages. Section 3.1 explains the feature extraction method and learning algorithm we used. Section 3.2 introduces the idea of using a belief network in search of the best global class labelling and the applied inference algorithm.

3.1 Learning Basic Classifiers

Although the labels of entities and relations from a sentence mutually depend on each other, two basic classifiers for entities and relations are first learned, in which a multi-class classifier for E(R) is learned as a function of all other "known" properties of the observation. The classifier for entities is a named entity classifier, in which the boundary of an entity is predefined (Collins and Singer, 1999). On the other hand, the relation classifier is given a pair of entities, which denote the two arguments of the target relation. Accurate predictions of these two classifiers seem to rely on complicated syntax analysis and semantics related information of the whole sentence. However, we derive weak classifiers by treating these two learning tasks as shallow text processing problems. This strategy has been successfully applied on several NLP tasks, such as information extraction (Califf and Mooney, 1999; Freitag, 2000; Roth and Yih, 2001) and chunking (i.e. shallow paring) (Munoz et al., 1999). It assumes that the class labels can be decided by local properties, such as the information provided by the words around or inside the target and their pos. Examples include the spelling of a word, part-of-speech, and semantic related attributes acquired from external resources such as WordNet.

The propositional learner we use is SNoW (Roth, 1998; Carleson et al., 1999), which can be downloaded from the web.[1] SNoW is a multi-class classifier that is specifically tailored for large scale learning tasks. Its learning architecture learns a sparse network of linear functions, in which the targets (entity classes or relation classes, in this case) are represented as linear functions over a common feature space. SNoW is built on a feature efficient learning algorithm, Winnow (Littlestone, 1988) that is suitable for learning in NLP-like domains, where the number of potential features is very large, but only a few of them are active in each example, and only a small fraction of them are relevant to the target concept.

While SNoW can be used as a classifier and predicts using a winner-take-all mechanism over the activation value of the target classes, here we rely directly on the raw activation value it outputs, which is the weighted linear sum of the features, to estimate the posteriors. It can be verified that the resulting values are monotonic with the confidence in the prediction, therefore is a good source of probability estimation. We use softmax (Bishop, 1995) over the raw activation values as probabilities. Specifically, suppose the number of classes is n, and the raw activation values of class i is act_i. The posterior estimation for class i is derived by the following equation.

$$p_i = \frac{e^{act_i}}{\sum_{1 \leq j \leq n} e^{act_j}}$$

[1] available at http://l2r.cs.uiuc.edu/∼cogcomp/cc-software.html

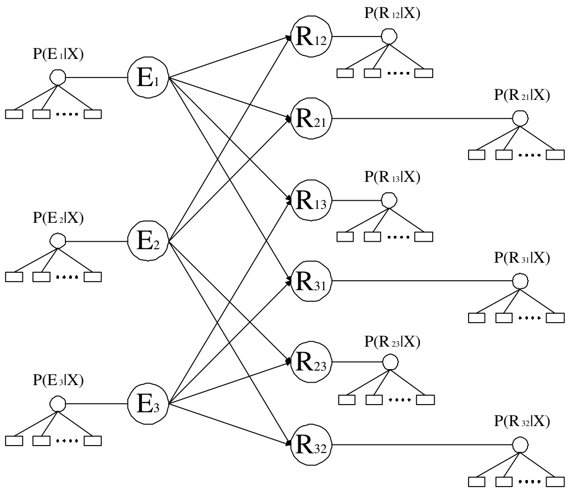

Figure 3: Belief network of 3 entity nodes and 6 relation nodes

3.2 Bayesian Inference Model

Broadly used in the AI community, *belief network* is a graphical representation of a probability distribution (Pearl, 1988). It is a directed acyclic graph (DAG), where the nodes are random variables and each node is associated with a conditional probability table which defines the probability given its parents. We construct a belief network that represents the constraints existing among R's and E's. Then, for each sentence, we use the classifiers from section 3.1 to compute the $Prob(E|observations)$ and $Prob(R|observations)$, and use the belief network to compute the most probable global predictions of the class labels.

The structure of our belief network, which represents the constraints is a bipartite graph. In particular, the variable E's and R's are the nodes in the network, where the E nodes are in one layer, and the R nodes are in the other. Since the label of a relation is dependent on the entity classes of its arguments, the links in the network connect the entity nodes, and the relation nodes that have these entities as arguments. For instance, node R_{ij} has two incoming links from nodes E_i and E_j. The conditional probabilities $P(R_{ij}|E_i, E_j)$ encodes the constraints as in Property 1. As an illustration, Figure 3 shows a belief network that consists of 3 entity nodes and 6 relation nodes.

Finding a most probable class assignment to the entities and relations is equivalent to finding the assignment of all the variables in the belief network that maximizes the joint probability. However, this most-probable-explanation (MPE) inference problem is intractable (Roth, 1996) if the network contains loops (undirected cycles), which is exactly the case in our network. Therefore, we resort to the following approximation method instead.

Recently, researchers have achieved great success in solving the problem of decoding messages through a noisy channel with the help of belief networks (Gallager, 1962; MacKay, 1999). The network structure used in their problem is similar to the network used here, namely a loopy bipartite DAG. The inference algorithm they used is Pearl's belief propagation algorithm (Pearl, 1988), which outputs exact posteriors in linear time if the network is singly connected (i.e. without loops) but does not guarantee to converge for loopy networks. However, researchers have empirically demonstrate that by iterating the belief propagation algorithm several times, the outputted values often converge to the right posteriors (Murphy et al., 1999). Due to the existence of loops, we also apply belief propagation algorithm iteratively as our inference procedure.

4 Experiments

The following subsections describe the data preparation process, the approaches tested in the experiments, and the experimental results.

4.1 Data Preparation

In order to build different datasets, we first collected sentences from TREC documents, which are mostly daily news such as Wall Street Journal, Associated Press, and San Jose Mercury News. Among the collected sentences, 245 sentences contain relation *kill* (i.e. two entities that have the *murder-victim* relation). 179 sentences contain relation *born_in* (i.e. a pair of entities where the second is the birthplace of the first). In addition to the above sentences, we also collected 502 sentences that contain no relations.[2]

Entities in these sentences are segmented by the simple rule: consecutive proper nouns and commas are combined and treated as an entity. Predefined entity class labels include *other_ent*, *person*, and *location*. Moreover, relations are defined by every pair of entities in a sentence, and the relation class labels defined are *other_rel*, *kill*, and *birthplace*.

Three datasets are constructed using the collected sentences. Dataset "kill" has all the 245 sentences of relation *kill*. Dataset "born_in" has all the 179 sentences of relation *born_in*. The third dataset "all" mixes all the sentences.

4.2 Tested Approaches

We compare three approaches in the experiments: *basic*, *omniscient*, and *BN*. The first approach, *basic*, tests our baseline – the performance of the basic classifiers. As described in Section 3.1, these classifiers are learned independently using local features and make predictions on entities and relations separately. Without taking global interactions into account, the features extracted are described as follows. For the entity classifier, features from the words around each entity are: words, tags, conjunctions of words and tags, bigram and trigram of words and

[2]The dataset will be made available on the web shortly.

tags. Features from the entity itself include the number of words it contains, bigrams of words in it, and some attributes of the words inside such as the prefix and suffix. In addition, whether the entity has some strings that match the names of famous people and places is also used as a feature. For the relation classifier, features are extracted from words around and between the two entity arguments. The types of features include bigrams, trigrams, words, tags, and words related "kill" and "birth" retrieved from WordNet.

The second approach, *omniscient*, is similar to *basic*. The only difference here is the labels of entities are revealed to the R classifier and vice versa. It is certainly impossible to know the true entity and relation labels in advance. However, this experiment may give us some ideas about how much the performance of the entity classifier can be enhanced by knowing whether the target is involved in some relations, and also how much the relation classifier can be benefited from knowing the entity labels of its arguments. In addition, it also provides a comparison to see how well the belief network inference model can improve the results.

The third approach, *BN*, tests the ability of making global inferences in our framework. We use the Bayes Net Toolbox for Matlab by Murphy [3] to implement the network and set the maximum number of the iteration of belief propagation algorithm as 20. Given the probabilities estimated by basic classifiers, the network infers the labels of the entities and relations globally in a sentence. Compared to the first two approaches, where some predictions may violate the constraints, the belief network model incorporates the constraints between entities and relations, thus all the predictions it makes will be coherent.

All the experiments of these approaches are done in 5-fold validation. In other words, these datasets are randomly separated into 5 disjoint subsets, and experiments are done 5 times by iteratively using 4 of them as training data and the rest as testing.

4.3 Results

The experimental results in terms of recall, precision,and $F_{\beta=1}$ for datasets "kill", "born_in", and "all" are given in Table 1, Table 2, and Table 3 respectively. We discuss two interesting facts of the results as follows.

First, the belief network approach tends to decrease recall in a small degree but increase precision significantly. This phenomenon is especially clear on the classification results of some relations. As a result, the F_1 value of the relation classification results is still enhanced to the extent that is near or even higher than the results of the *Omniscient* approach. This may be explained by the fact that if the label of a relation is predicted as positive (i.e. not *other_rel*), the types of its entity arguments must satisfy the constraints. This inference process reduces the number of false positive, thus enhance the precision.

[3] available at http://www.cs.berkeley.edu/∼murphyk/Bayes/bnt.html

Second, knowing the class labels of relations does not seem to help the entity classifier much. In all three datasets, the difference of *Basic* and *Omniscient* approaches is usually less than 3% in terms of F_1, which is not very significant given the size of our datasets. This phenomenon may be due to the fact that only a few of entities in a sentence are involved in some relations. Therefore, it is unlikely that the entity classifier can use the relation information to correct its prediction.

Approach	person			location		
	Rec	Prec	F_1	Rec	Prec	F_1
Basic	96.6	92.3	94.4	76.3	91.9	83.1
BN	89.0	96.1	92.4	78.8	86.3	82.1
Omniscient	96.4	92.6	94.5	75.4	90.2	81.9
Approach	**kill**					
	Rec	Prec	F_1			
Basic	61.8	57.2	58.6			
BN	49.8	85.4	62.2			
Omniscient	67.7	63.6	64.8			

Table 1: Results for dataset "kill"

Approach	person			location		
	Rec	Prec	F_1	Rec	Prec	F_1
Basic	85.5	90.7	87.8	89.5	93.2	91.1
BN	87.0	90.9	88.8	87.5	93.4	90.3
Omniscient	90.6	93.4	91.7	90.7	96.5	93.4
Approach	**born_in**					
	Rec	Prec	F_1			
Basic	81.4	63.4	70.9			
BN	87.6	70.7	78.0			
Omniscient	86.9	71.8	78.0			

Table 2: Results for dataset "born_in"

Approach	person			location		
	Rec	Prec	F_1	Rec	Prec	F_1
Basic	92.1	87.0	89.4	83.2	81.1	82.0
BN	78.8	94.7	86.0	83.0	81.3	82.1
Omniscient	93.4	87.3	90.2	83.5	83.1	83.2
Approach	**kill**			**born_in**		
	Rec	Prec	F_1	Rec	Prec	F_1
Basic	43.8	78.6	55.0	69.0	72.9	70.5
BN	47.2	86.8	60.7	68.4	87.5	76.6
Omniscient	52.8	79.5	62.1	76.1	71.3	73.2

Table 3: Results for dataset "all"

5 Discussion

The promising results of our preliminary experiments demonstrate the feasibility of our probabilistic framework. For the future work, we plan to extend this research in the following directions.

The first direction we would like to explore is to apply our framework in a boot-strapping manner. The main difficulty in applying learning on NLP problems is not lack of text corpus, but lack of *labelled* data. Bootstrapping, applying the classifiers to autonomously annotate the data and using the new data to train and improve existing classifiers, is a promising approach. Since the precision of our framework is pretty high, it seems possible to use the global inference to annotate new data. Based on this property, we can derive an EM-like approach for labelling and inferring the types of entities and relations simultaneously. The basic idea is to use the global inference output as a mean to annotate entities and relations. The new annotated data can then be used to train classifiers, and the whole process is repeated again.

The second direction is to improve our probabilistic inference model in several ways. First, since the results of the inference procedure we use, the loopy belief propagation algorithm, produces approximate values, some of the results can be wrong. Although the computational time of the exact inference algorithm for loopy network is exponential, we may still be able to run it given the small number of variables that are of interest each time in our case. Therefore, we can further check if performance suffers from the approximation. Second, the belief network model may not be expressive enough since it allows no cycles. To fully model the problem, cycles may be needed. For example, the class labels of R_{12} and R_{21} actually depend on each other. (e.g. If R_{12} is *born_in*, then R_{21} will not be *born_in* or *kill*.) Similarly, the class labels of E_1 and E_2 can depend on the labels of R_{12}. To fully represent the mutual dependencies, we would like to explore other probabilistic models that are more expressive than the belief network.

References

S. P. Abney. 1991. Parsing by chunks. In S. P. Abney R. C. Berwick and C. Tenny, editors, *Principle-based parsing: Computation and Psycholinguistics*, pages 257–278. Kluwer, Dordrecht.

C. Bishop, 1995. *Neural Networks for Pattern Recognition*, chapter 6.4: Modelling conditional distributions, page 215. Oxford University Press.

M. Califf and R. Mooney. 1999. Relational learning of pattern-match rules for information extraction. In *National Conference on Artificial Intelligence*.

A. Carleson, C. Cumby, J. Rosen, and D. Roth. 1999. The SNoW learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department, May.

M. Collins and Y. Singer. 1999. Unsupervised models for name entity classification. In *EMNLP-VLC'99, the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, June.

D. Freitag. 2000. Machine learning for information extraction in informal domains. *Machine Learning*, 39(2/3):169–202.

R. Gallager. 1962. Low density parity check codes. *IRE Trans. Info. Theory*, IT-8:21–28, Jan.

L. Hirschman, M. Light, E. Breck, and J. Burger. 1999. Deep read: A reading comprehension system. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the International Conference on Machine Learning*.

N. Littlestone. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318.

D. MacKay. 1999. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45.

M. Munoz, V. Punyakanok, D. Roth, and D. Zimak. 1999. A learning approach to shallow parsing. In *EMNLP-VLC'99, the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, June.

K. Murphy, Y. Weiss, and M. Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proc. of Uncertainty in AI*, pages 467–475.

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.

V. Punyakanok and D. Roth. 2001. The use of classifiers in sequential inference. In *NIPS-13; The 2000 Conference on Advances in Neural Information Processing Systems*.

D. Roth and W. Yih. 2001. Relational learning via propositional algorithms: An information extraction case study. In *Proc. of the International Joint Conference on Artificial Intelligence*, pages 1257–1263.

D. Roth. 1996. On the hardness of approximate reasoning. *Artificial Inteligence*, 82(1-2):273–302, April.

D. Roth. 1998. Learning to resolve natural language ambiguities: A unified approach. In *Proc. National Conference on Artificial Intelligence*, pages 806–813.

E. Voorhees. 2000. Overview of the trec-9 question answering track. In *The Ninth Text Retrieval Conference (TREC-9)*, pages 71–80. NIST SP 500-249.