Appears in *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-99).*

Constructing Biological Knowledge Bases by Extracting Information from Text Sources

Mark Craven and **Johan Kumlien**
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, Pennsylvania, 15213-3891, U.S.A.
mark.craven@cs.cmu.edu johan.kumlien@cs.cmu.edu

Abstract

Recently, there has been much effort in making databases for molecular biology more accessible and interoperable. However, information in text form, such as MEDLINE records, remains a greatly underutilized source of biological information. We have begun a research effort aimed at automatically mapping information from text sources into structured representations, such as knowledge bases. Our approach to this task is to use machine-learning methods to induce routines for extracting facts from text. We describe two learning methods that we have applied to this task — a statistical text classification method, and a relational learning method — and our initial experiments in learning such information-extraction routines. We also present an approach to decreasing the cost of learning information-extraction routines by learning from "weakly" labeled training data.

Introduction

The science of molecular biology has been greatly affected by the proliferation of the Internet in recent years. There are now hundreds of on-line databases characterizing biological information such as sequences, structures, molecular interactions and expression patterns. Moreover, there are servers that perform such tasks as identifying genes in DNA sequences (e.g. GRAIL, Xu *et al.*, 1996) and predicting protein secondary structures (e.g. PredictProtein, Rost, 1996). And there are systems that integrate information from various sources (e.g. The Genome Channel, Genome Annotation Consortium, 1999), provide interoperability among distributed databases (e.g. Entrez, National Center for Biotechnology Information, 1999) and support knowledge-based reasoning (e.g. EcoCyc, Karp *et al.*, 1997). Another rich source of on-line information is the scientific literature. The MEDLINE database, for example, provides bibliographic information and abstracts for more than nine million articles that have been published in biomedical journals. A fundamental limitation of MEDLINE and similar sources, however, is that the information they contain is not represented in structured format, but instead in natural language text. The goal of our research is to develop methods that can inexpensively and accurately map information in scientific text sources, such as MEDLINE, into a structured representation, such as a knowledge base or a database. Toward this end, we have developed novel methods for automatically extracting key facts from scientific texts.

Current systems for accessing MEDLINE (e.g. Pubmed, National Library of Medicine, 1999a) accept keyword-based queries to text sources and return documents that are (hopefully) relevant to the query. Our goal, in contrast, is to support the kinds of arbitrarily complex queries that current database systems handle, and to return actual answers rather than relevant documents. The system we are developing is motivated by several different types of tasks that we believe would greatly benefit from the ability to extracted structured information from text:

- **Database construction and updating.** Our system could be used to help construct and update databases and knowledge bases by extracting fields from text. For example, we are currently working with a team that is developing a knowledge base of protein localization patterns (Boland, Markey, & Murphy 1996). We are using our system to assist in developing an ontology of localization patterns and to populate the database with text-extracted facts describing the localization patterns of individual proteins. In a similar vein, our system could be used to update databases that track particular classes of mutation studies (Lathrop *et al.* 1998), and to provide automatic genome annotation for a system such as The Genome Channel (Genome Annotation Consortium 1999) or EcoCyc (Karp *et al.* 1997).
- **Summarization.** Another promising application of our system is to provide structured summaries of what is known about particular biological objects. For example, we are working with scientists who are studying the genetic basis of diseases by identifying expressed sequence tags that are differentially expressed in tissues in various states. Frequently, these scientists do time-consuming MEDLINE searches to

determine if some candidate gene product is likely to be related to the disease of interest. When performing these searches, the scientists typically are trying to answer such questions as: In what types of tissues, cells and subcellular locations is the protein known to be expressed? Is the protein known to be associated with any diseases? Is the protein known to interact with any pharmacological agents? We plan to partially automate the task of extracting answers to these questions from text.

- **Discovery.** An especially compelling application of our system is its potential application to scientific discovery. The articles in MEDLINE describe a vast web of relationships among the genes, proteins, pathways, tissues and diseases of various systems and organisms of interest. Moreover, each article describes only a small piece of this web. The work of Swanson *et al.* (Swanson & Smalheiser 1997) has demonstrated that significant but previously unknown relationships among entities (e.g. magnesium and migraine headaches) can be discovered by automatically eliciting this information from the literature. Swanson's algorithm detects relationships among objects simply by considering the statistics of word co-occurrences in article titles. We conjecture that such relationships can be detected more accurately by our method of analyzing sentences in the article's abstract or text. Moreover, whereas Swanson's algorithm posits only that *some* relation holds between a pair of objects, our system is designed to state what the specific relation is.

One conceivable approach to devising a system to solve tasks such as these would be to perform full natural language understanding of the text. This undertaking, however, is well beyond the capabilities of current natural language systems. Our approach is to treat the task as one of *information extraction.* Information extraction (IE) involves a limited form of natural language processing in which the system tries only to extract predefined classes of facts from the text. A key aspect of our approach is that we use machine-learning algorithms to induce our information extractors.

In the following section, we describe the information-extraction task in more detail. We then describe a statistical text-classification approach to learning information extractors, and present an empirical evaluation of this method. A key limitation of using machine-learning methods to induce information-extraction methods is that the process of labeling training examples is expensive. The fourth section of the paper presents an approach to learning information extractors that exploits existing databases to automatically label training examples. The promise of this approach is that it can greatly reduce the cost of assembling sets of labeled training data. We then present a second approach to learning information extractors that exploits more linguistic knowledge than our initial approach. Finally, we discuss related work, the contributions and limitations of our work, and some directions we are pursuing in our current research.

The Information Extraction Task

The general information extraction task can be formulated as follows:

> **Given:** (i) a set of classes of interest and relations among these classes, and (ii) a corpus of documents to be processed.
> **Do:** extract from the documents instances of the classes and relations that are described in the documents.

This limited form of natural language understanding has been the focus of much research over the past decade (Cowie & Lehnert 1996; Cardie 1997). Most of the work in this community has involved hand-coding extraction routines. However, in recent years there have been several research efforts investigating the application of machine learning methods to inducing information extractors (Riloff 1996; Soderland 1996; Califf 1998; Freitag 1998; Soderland 1999). Machine learning methods offer a promising alternative to hand coding IE routines because they can greatly reduce the amount of time and effort required to develop such methods.

In the applications we are addressing, we are primarily interested in extracting instances of *relations* among objects. In particular, we want to learn extractors for the following:[1]

- subcellular-localization(Protein, Subcellular-Structure): the instances of this relation represent proteins and the subcellular structures in which they are found.
- cell-localization(Protein, Cell-Type): the cell types in which a given protein is found.
- tissue-localization(Protein, Tissue): the tissue types in which a given protein is found.
- associated-diseases(Protein, Disease): the diseases with which a given protein is known to have some association.
- drug-interactions(Protein, Pharmacologic-Agent): the pharmacologic agents with which a given protein is known to interact.

In our initial experiments we are focusing on the subcellular-localization relation. As an example of the IE task, Figure 1 shows several sentences and the instances of the subcellular-localization relation that we would like to extract from them.

Extraction via Text Classification

Our first approach to learning information extractors uses a statistical text classification method. Without loss of generality, assume that we are addressing the

[1] We use the following notation to describe relations: constants, such as the names of specific relations and the objects they characterize, start with lowercase letters; the names of variables begin with uppercase letters.

Immunoprecipitation of biotinylated type XIII collagen from surface-labeled HT-1080 cells, subcellular fractionation, and immunofluorescence staining were used to demonstrate that type XIII collagen molecules are indeed located in the plasma membranes of these cells.	subcellular-localization(collagen, plasma-membranes)
HSP47 is a collagen-binding stress protein and is thought to be a collagen-specific molecular chaperone, which plays a pivotal role during the biosynthesis and secretion of collagen molecules in the endoplasmic reticulum.	subcellular-localization(collagen, endoplasmic-reticulum)

Figure 1: An illustration of the IE task. On the left are sentences from MEDLINE abstracts. On the right are instances of the subcellular-localization relation that we might extract from these sentences.

task of extracting instances of a binary relation, r(X, Y). This approach assumes that for the variables of the relation, X and Y, we are given semantic lexicons, L(X) and L(Y), of the possible words that could be used in instances of r. For example, the second constant of each instance of the relation subcellular-localization, described in the previous section, is in the semantic class Subcellular-Structure. Our semantic lexicon for this class consists of words like *nucleus, mitochondrion*[2], etc. Given such lexicons, the first step in this approach is to identify the *instances* in a document that could possibly express the relation. In the work reported here, we make the assumption that these instances consist of individual sentences. Thus, we can frame the information-extraction task as one of sentence classification. We extract a relation instance r(x, y) from the sentence if (i) the sentence contains a word x $\in$ L(X) and a word y $\in$ L(Y), and (ii) the sentence is classified as a positive instance by a statistical model. Otherwise, we consider the sentence to be a negative instance and we do not extract anything from it. We can learn the statistical model used for classification from labeled positive and negative instances (i.e. sentences that describe instances of the relation, and sentences that do not).

As stated above, we make the assumption that instances consist of individual sentences. It would be possible, however, to define instances to be larger chunks of text (e.g. paragraphs) or smaller ones (e.g. sentence clauses) instead. One limitation of this approach is that it forces us to assign only one class label to each instance. Consider, for example, a sentence that mentions multiple proteins and multiple subcellular locations. The sentence may specify that only some of these proteins are found in only some of the locations. However, we can only classify the sentence as being a member of the positive class, in which case we extract all protein/location pairs as instances of the target relation, or we classify the sentence as a negative instance, in which case we extract no relation instances from the sentence. This limitation provides an argument for setting up the task so that instances are relatively small.

[2]Our lexicons also include adjectives and the plural forms of nouns.

In order to learn models for classifying sentences, we use a statistical text-classification method. Specifically, we use a Naive Bayes classifier with a *bag-of-words* representation (Mitchell 1997). This approach involves representing each document (i.e. sentence) as a bag of words. The key assumption made by the bag-of-words representation is that the position of a word in a document does not matter (e.g. encountering the word *protein* at the beginning of a document is the same as encountering it at the end).

Given a document d of n words $(w_1, w_2, \ldots, w_n)$, Naive Bayes estimates the probability that the document belongs to each possible class $c_j \in C$ as follows:

$$\Pr(c_j|d) = \frac{\Pr(c_j)\Pr(d|c_j)}{\Pr(d)} \approx \frac{\Pr(c_j)\prod_{i=1}^{n}\Pr(w_i|c_j)}{\Pr(d)}. \quad (1)$$

In addition to the position-independence assumption implicit in the bag-of-words representation, Naive Bayes makes the assumption that the occurrence of a given word in a document is independent of all other words in the document. Clearly, this assumption does not hold in real text documents. However, in practice, Naive Bayes classifiers often perform quite well (Domingos & Pazzani 1997; Lewis & Ringuette 1994).

The prior probability of the document, $\Pr(d)$ does not need to be estimated directly. Instead we can get the denominator by normalizing over all of the classes. The conditional probability, $\Pr(w_i|c_j)$, of seeing word w_i given class c_j is estimated from the training data. In order to make these estimates robust with respect to infrequently encountered words, we use Laplace estimates:

$$\Pr(w_i|c_j) = \frac{N(w_i, c_j) + 1}{N(c_j) + T}, \quad (2)$$

where $N(w_i, c_j)$ is the number of times word w_i appears in training set examples from class c_j, $N(c_j)$ is the total number of words in the training set for class c_j and T is the total number of unique words in the training set.

Before applying Naive Bayes to our documents, we first preprocess them by *stemming* words. Stemming refers to the process of heuristically reducing words to their root form (Porter 1980). For example the words *localize*, *localized* and *localization* would be stemmed to the root *local*. The motivation for this step is to make commonalities in related sentences more apparent to the learner.

To evaluate our approach, we assembled a corpus of abstracts from the MEDLINE database. This corpus, consisting of 2,889 abstracts, was collected by querying on the names of six proteins and then downloading the first 500 articles returned for each query protein, discarding entries that did not include an abstract. We selected the six proteins for their diversity and for their relevance to the research of one of our collaborators. The six proteins/polypeptides are: serotonin (a neurotransmitter), secretin (a hormone), NMDA receptor (a receptor), collagen (a structural protein), trypsinogen (an enzyme), and calcium channel (an ion channel).

We created a labeled data set for our IE experiments as follows. One of us (Kumlien), who is trained in medicine and clinical chemistry, hand-annotated each abstract in the corpus with instances of the target relation subcellular-localization. To determine if an abstract should be annotated with a given instance, subcellular-localization(x, y), the abstract had to clearly indicate that protein x is found in location y. To aid in this labeling process, we wrote software that searched the abstracts for words from the location lexicon, and suggested candidate instances based on search hits. This labeling process resulted in a total of thirty-three instances of the subcellular-localization relation. Individual instances were found in from one to thirty different abstracts. For example, the fact that calcium channels are found in the sarcoplasmic reticulum was indicated in eight different abstracts.

The goal of the information-extraction task is to correctly identify the instances of the target relation that are represented in the corpus, without predicting spurious instances. Furthermore, although each instance of the target relation, such as subcellular-localization(calcium-channels, sarcoplasmic-reticulum), may be represented multiple times in the corpus, we consider the information-extraction method to be correct as long it extracts this instance from *one* of its occurrences. We estimate the accuracy of our learned sentence classifiers using leave-one-out cross validation. Thus, for every sentence in the data set, we induce a classifier using the other sentences as training data, and then treat the held-out sentence as a test case. We compare our learned information extractors against a baseline method that we refer to as the *sentence co-occurrence* predictor. This method predicts that a relation holds if a protein and a sub-cellular location occur in the same sentence.

We consider using our learned Naive Bayes models in two ways. In the first method, we use them as classifiers: given an instance, the model either classifies it as positive and returns an extracted relation instance, or the model classifies it as negative and extracts nothing. To use Naive Bayes for classification, we simply return the most probable class. In the second method, the model returns its estimated posterior probability that the instance is positive. With this method, we do not strictly accept or reject sentences.

For each method, we rank its predictions by a confidence measure. For a given relation instance, r(x, y), we first collect the set of sentences that would assert this relation if classified into the positive class (i.e. those sentences that contain both the term x and the term y). For the sentence co-occurrence predictor, we rank a predicted relation instance by the size of this set. When we use the Naive Bayes models as classifiers, we rank a predicted relation instance by the number of sentences in this set that are classified as belonging to the positive class. In the second method, where we use the probabilities produced by Naive Bayes, we estimate the posterior probability that each sentence is in the positive class and combine the class probabilities using the *noisy or* function (Pearl 1988):

$$\text{confidence} = 1 - \prod_{k}^{N} \left[1 - \Pr(c = \text{pos} \mid s_k)\right].$$

Here, $\Pr(c = \text{pos} \mid s_k)$ is the probability estimated by Naive Bayes for the kth element of our set of sentences. This combination function assumes that each sentence in the set provides independent evidence for the truth of the asserted relation.

Since we have a way to rank the predictions produced by each of our methods, we can see how the accuracy of their predictions vary with confidence. Figure 2 plots *precision* versus *recall* for the three methods on the task of extracting instances of the subcellular-localization relation. Precision and recall are defined as follows:

$$\text{precision} = \frac{\#\ \text{correct positive predictions}}{\#\ \text{positive predictions}},$$

$$\text{recall} = \frac{\#\ \text{correct positive predictions}}{\#\ \text{positive instances}}.$$

Figure 2 illustrates several interesting results. The most significant result is that both versions of the Naive Bayes predictor generally achieve higher levels of precision than the sentence co-occurrence predictor. For example, at 25% recall, the precision of the baseline predictor is 44%, whereas for the Naive Bayes classifiers it is 70%, and for the Naive Bayes models using noisy-or combination it is 62%. This result indicates that the learning algorithm has captured some of the statistical regularities that arise in how authors describe the subcellular localization of a protein. None of the methods is able to achieve 100% recall since some positive relation instances are not represented by individual sentences. In the limit, the recall of the Naive Bayes classifiers is not as high as it is for the baseline predictor because the former incorrectly classifies as negative some sentences representing positive instances. Since the Naive Bayes models with noisy-or do not reject any sentences in this way, their recall is the same as the baseline method. Their precision is lower than the Naive Bayes classifier,

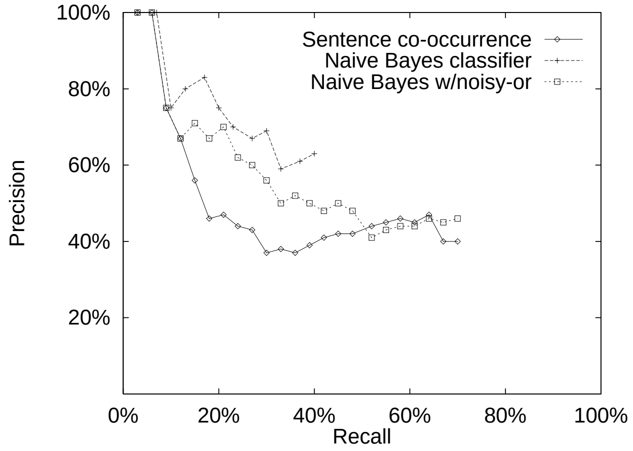

Figure 2: Precision vs. recall for the co-occurrence predictors and the Naive Bayes model.

however, indicating that even when Naive Bayes makes accurate classifications, it often does not estimate probabilities well (Domingos & Pazzani 1997). An interesting possibility would be to combine these predictors to get the high precision of the Naive Bayes classifiers along with the high recall of the Naive Bayes models using noisy-or. Provost and Fawcett (1998) have developed a method especially well suited to this type of combination.

Exploiting Existing Databases for Training Data

We have argued that machine learning offers a promising alternative to hand-coding information extraction routines because the hand-coding process has proven to be so time-consuming. A limitation of the machine-learning approach, however, is that providing labeled training data to the learner is itself quite time-consuming and tedious. In fact, labeling the corpus used in the previous section required approximately 35 hours of an expert's time. In this section, we present an approach to learning information extractors that relies on existing databases to provide something akin to labeled training instances.

Our approach is motivated by the observation that, for many IE tasks, there are existing information sources (knowledge bases, databases, or even simple lists or tables) that can be coupled with documents to provide what we term "weakly" labeled training examples. We call this form of training data weakly labeled because each instance consists not of a precisely marked document, but instead it consists of a fact to be extracted along with a document that *may* assert the fact. To make this concept more concrete, consider the Yeast Protein Database (YPD) (Hodges, Payne, & Garrels 1998), which includes a *subcellular localization* field for many proteins. Moreover, in some cases the entry for this field has a reference (and a hyperlink to the PubMed entry for the reference) to the article that established the subcellular localization fact. Thus, each of these entries along with its reference could be used as a weakly labeled instance for learning our subcellular-localization information extractors.

In this section we evaluate the utility of learning from weakly labeled training instances. From the YPD Web site, we collected 1,213 instances of the subcellular-localization relation that are asserted in the YPD database, and from PubMed we collected the abstracts from 924 articles that are pointed to by these entries in YPD. For many of the relation instances, the associated abstracts do not say anything about the subcellular localization of the reference protein, and thus they are not helpful to us. However, if we select the relation instances for which an associated abstract contains a sentence that mentions both the protein and a subcellular location, then we wind up with 336 relation instances described in 633 sentences. This data set contains significantly more relation instances than the one we obtained via hand-labeling, and it was acquired by a completely automated process.

As in the previous section, we treat individual sentences as instances to be processed by a Naive Bayes text classifier. Moreover, we make the assumption that every one of the 633 sentences mentioned above represents a positive training example for our text classifier. In other words, we assume that if we know that relation subcellular-localization(x, y) holds, then any sentence in the abstract(s) associated with subcellular-localization(x, y) that references both x and y is effectively stating that x is located in y. Of course this assumption is not always valid in practice. We take the remaining sentences in the YPD corpus as negative training examples.

The hypothesis that we consider in this section is that it is possible to learn accurate information-extraction routines using weakly labeled training data, such as that we gathered from YPD. To test this hypothesis we train a Naive Bayes model using the YPD data as a training set, and then we evaluate it using our hand-labeled corpus as a test set. We train our statistical text classifier in the same manner as described in the previous section.

Figure 3 shows the precision vs. recall curves for this experiment. As a baseline, the figure also shows the precision/recall curve for the sentence co-occurrence predictor described in the previous section. Recall that the co-occurrence predictor does not use a training set in any way; it simply makes its predictions by noting co-occurrence statistics in the test set. Therefore, it is an appropriate baseline no matter what training set we use.

From this figure we can see that the Naive Bayes model learned from the YPD curve is comparable to the curve for the models learned from the hand-labeled data. Whereas the Naive Bayes classifiers from the previous section achieved 69% precision at 30% recall, the Naive Bayes classifier trained on the YPD data reaches 77% precision at 30% recall. Moreover, the YPD model

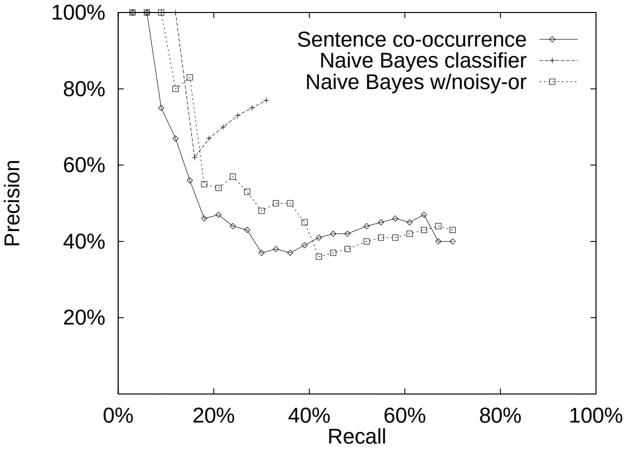

Figure 3: Precision vs. recall for the Naive Bayes model trained on the YPD data set.

achieves better precision at comparable levels of recall than the sentence co-occurrence classifier.

These two results support our hypothesis. It should be emphasized that the result of this experiment was not a foregone conclusion. Although the YPD data set contains many more positive instances than our hand-labeled data set, this data set represents a very different distribution of text than our test set. The YPD data set has a particular focus on the localization of yeast proteins. The test set, in contrast does not concentrate on protein localization and barely mentions yeast. We argue that the result of this experiment is very significant result because it indicates that effective information-extraction routines can be learned without an expensive hand-coding or hand-labeling process.

One way to obtain insight into our learned text classifiers is to ask which words contribute most highly to the quantity $\Pr(\text{pos}|d)$ (i.e. the predicted probability that a document d belongs to the positive class). To measure this, we calculate

$$\log\left(\frac{\Pr(w_i|\text{pos})}{\Pr(w_i|\text{neg})}\right) \tag{3}$$

for each word w_i in the vocabulary of the model learned from the YPD data set. Figure 4 shows the twenty stemmed words, excluding words that refer to specific subcellular locations, that have the greatest value of this log-odds ratio. The vocabulary for this learned model includes more than 2500 stemmed words. As the table illustrates, many of the highly weighted words are intuitively natural predictors of sentences that describe subcellular-localization facts. The words in this set include *local, insid, immunofluoresc, immunoloc, accumul,* and *microscopi*. Some of the highly weighted words, however, are not closely associated with the concept of subcellular localization. Instead, their relatively large weights simply reflect the fact that it is difficult to reliably estimate such probabilities from limited training data.

stemmed word	$\log\left(\frac{\Pr(w_i\|\text{pos})}{\Pr(w_i\|\text{neg})}\right)$
local	0.00571
pmr	0.00306
dpap	0.00259
insid	0.00209
indirect	0.00191
galactosidas	0.00190
immunofluoresc	0.00182
secretion	0.00181
mcm	0.00157
mannosidas	0.00157
sla	0.00156
gdpase	0.00156
bafilomycin	0.00154
marker	0.00141
presequ	0.00125
immunoloc	0.00125
snc	0.00121
stain	0.00115
accumul	0.00114
microscopi	0.00112

Figure 4: The twenty stemmed words (aside from words referring to specific subcellular locations) weighted most highly by the YPD-trained text classifier. The weights represent the log-odds ratio of the words given the positive class.

Extraction via Relational Learning

The primary limitation of the statistical classification approach to IE presented in the preceding sections is that it does not represent the linguistic structure of the text being analyzed. In deciding whether a given sentence encodes an instance of the target relation or not, the statistical text classifiers consider only what words occur in the sentence – not their relationships to one another. Surely, the grammatical structure of the sentence is important for our task, however.

To learn information extractors that are able to represent grammatical structure, we have begun exploring an approach that involves parsing sentences, and learning relational rules in terms of these parses. Our approach uses a sentence analyzer called Sundance (Riloff 1998) that assigns part-of-speech tags to words, and then builds a shallow parse tree that segments sentences into clauses and noun, verb, or prepositional phrases. Figure 5 shows the parse tree built by Sundance for one sentence in our corpus. The numbers shown in brackets next to the root and each phrase in the tree are identifiers that we can use to refer to a particular sentence in the corpus or to a particular phrase in a sentence.

Given these parses, we learn information-extraction rules using a relational learning algorithm that is similar to FOIL (Quinlan 1990). The appeal of using a relational method for this task is that it can naturally represent *relationships* among sentence constituents in

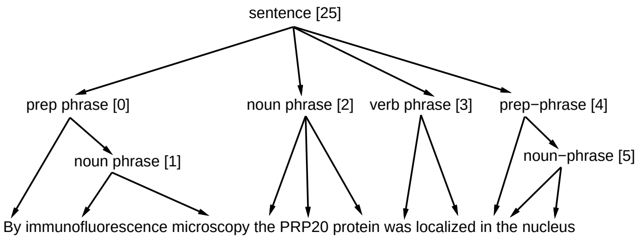

Figure 5: A parse tree produced by Sundance for one sentence in our YPD corpus.

learned rules, and it can represent an arbitrary amount of context around the parts of the sentence to be extracted.

The objective of the learning algorithm is to learn a definition for the predicate:
localization-sentence(Sentence-ID,Phrase-ID,Phrase-ID). Each instance of this relation consists of (i) an identifier corresponding to the sentence represented by the instance, (ii) an identifier representing the phrase in the sentence that contains an entry in the protein lexicon, and (iii) and identifier representing the phrase in the sentence that contains an entry in the subcellular location lexicon. Thus, the learning task is to recognize pairs of phrases that correspond to positive instances of the target relation. The models learned by the relational learner consist of logical rules constructed from the following *background relations*:

- phrase-type(Phrase-ID, Phrase-Type): This relation allows a particular phrase to be characterized as a noun phrase, verb phrase, or prepositional phrase.
- next-phrase(Phrase-ID, Phrase-ID): This relation specifies the order of phrases in a sentence. Each instance of the relation indicates the successor of one particular phrase.
- constituent-phrase(Phrase-ID, Phrase-ID): This relation indicates cases where one phrase is a constituent of another phrase. For example, in Figure 5, the first prepositional phrase in the sentence has a constituent noun phrase.
- subject-verb(Phrase-ID, Phrase-ID), verb-direct-object(Phrase-ID,Phrase-ID): These relations enable the learner to link subject noun phrases to their corresponding verb phrases, and verb phrases to their corresponding direct object phrases.
- same-clause(Phrase-ID, Phrase-ID): This relation links phrases that occur in the same sentence clause.

Training and test examples are described by instances of these relations. For example, Figure 6 shows the instances of the background and target relations that represent the parse tree shown in Figure 5. The constants used to represent the sentence and its phrases in Figure 6 correspond to the identifiers shown in brackets in Figure 5.

```
phrase-type(phrase-0, prepositional-phrase).
phrase-type(phrase-1, noun-phrase).
phrase-type(phrase-2, noun-phrase).
phrase-type(phrase-3, verb-phrase).
phrase-type(phrase-4, prepositional-phrase).
phrase-type(phrase-5, noun-phrase).

next-phrase(phrase-0, phrase-2).
next-phrase(phrase-2, phrase-3).
next-phrase(phrase-3, phrase-4).

constituent-phrase(phrase-0, phrase-1).
constituent-phrase(phrase-4, phrase-5).

subject-verb(phrase-2, phrase-3).

localization-sentence(sentence-25, phrase-2, phrase-5).
```

Figure 6: Our relational representation of the parse shown in Figure 5.

This set of background relations enables the learner to characterize the relations among phrases in sentences. Additionally, we also allow the learner to characterize the words in sentences and phrases. One approach to doing this would be to include another background relation whose instances linked individual words to the phrases and sentences in which they occur. We have investigated this approach and found that the learned rules often have low precision and/or recall because they are too dependent on the presence of particular words. The approach we use instead allows the learning algorithm to use Naive Bayes classifiers to characterize the words in sentences and phrases.

Figure 7 shows a rule learned by our relational method. The rule is satisfied when all of the literals to the right of the ":-" are satisfied. The first two literals specify that the rule is looking for sentences in which the phrase referencing the subcellular location follows the phrase referencing the protein, and there is one phrase separating them. The next literal specifies that the sentence must satisfy (i.e. be classified as positive by) a particular Naive Bayes classifier. The fourth literal indicates that the phrase referencing the protein must satisfy a Naive Bayes classifier. The two final literals specify a similar condition for the phrase referencing the subcellular location. The bottom part of Figure 7 shows the stemmed words that are weighted most highly by each of the naive Bayes classifiers.

Although the Naive Bayes predicates used in the rule shown in Figure 7 appear to overlap somewhat, their differences are noticeable. For example, whereas the predicate that is applied to the Protein-Phrase highly weights the words *protein*, *gene* and *product*, the predicates that are applied to the Location-Phrase focus on subcellular locations and prepositions such as *in*, *to* and *with*.

```
localization-sentence(Sentence, Protein-Phrase, Location-Phrase) :-
                next-phrase(Protein-Phrase, Phrase-1),
                next-phrase(Phrase-1, Location-Phrase),
                sentence-naive-bayes-1(Sentence),
                phrase-naive-bayes-1(Protein-Phrase),
                phrase-naive-bayes-2(Location-Phrase),
                phrase-naive-bayes-3(Location-Phrase).
```

sentence-naive-bayes-1:	nucleu, mannosidas, bifunct, local, galactosidas, nuclei, immunofluoresc, ...
phrase-naive-bayes-1:	protein, beta, galactosidas, gene, alpha, mannosidas, bifunct, product, ...
phrase-naive-bayes-2:	nucleu, nuclei, mitochondria, vacuol, plasma, insid, membran, atpas, ...
phrase-naive-bayes-3:	the, nucleu, in, mitochondria, membran, nuclei, to, vacuol, yeast, with, ...

Figure 7: Top: a rule learned by our relational method. This rule includes four Naive Bayes predicates. Bottom: the most highly weighted words (using the log-odds ratio) in each of the Naive Bayes predicates.

Using a procedure similar to *relational pathfinding* (Richards & Mooney 1992), our learning algorithm initializes each rule by trying to find the combination of next-phrase, constituent-phrase, subject-verb, verb-direct-object, and same-clause literals that link the phrases of the most uncovered positive instances. After the rule is initialized with these literals, the learning algorithm uses a hill-climbing search to add additional literals. The algorithm can either add a literal expressed using one of the background relations, or it can invent a new Naive Bayes classifier to characterize one of the phrases in the sentence or the sentence itself. This method for inventing Naive Bayes classifiers in the context of relational learning is described in detail elsewhere (Slattery & Craven 1998).

To evaluate our relational IE approach, we learned a set of rules using the YPD data set as a training set, and tested the rules on the hand-labeled data set. Our relational algorithm learned a total of 26 rules covering the positive instances in the training set.

Figure 8 shows the precision vs. recall curve for the learned relational rules. The confidence measure for a given example is the estimated accuracy of the first rule that the example satisfies. We estimate the accuracy of each of our learned rules by calculating an m-estimate (Cestnik 1990) of the rule's accuracy over the training examples. The m-estimate of a rule's accuracy is defined as follows:

$$m\text{-estimate accuracy} = \frac{n_c + mp}{n + m}$$

where n_c is the number of instances correctly classified by the rule, n is the total number of instances classified by the rule, p is a prior estimate of the rule's accuracy, and m is a constant called the *equivalent sample size* which determines how heavily p is weighted relative to the observed data. In our experiments, we set $m = 5$ and we set p to the proportion of instances in the training set that belong to the target class. We then use these m-estimates to sort the rules in order of descending estimated accuracy.

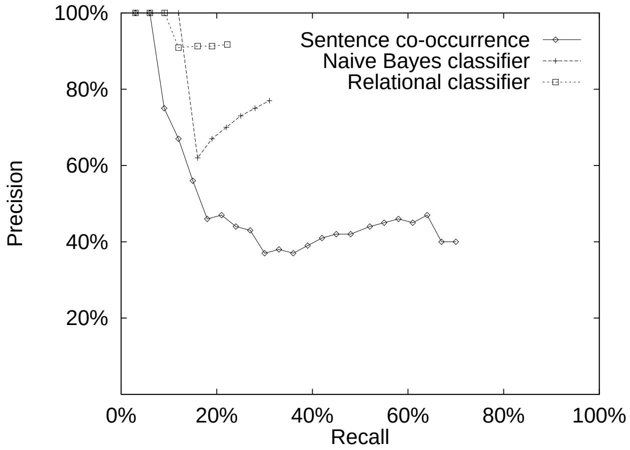

Figure 8: Precision vs. recall for the relational classifier trained on the YPD data set.

For comparison, Figure 8 also shows the precision vs. recall curves for the YPD-trained Naive Bayes classifier discussed in the previous section, and for the sentence co-occurrence baseline. As this figure illustrates, although the recall of the relational rule set is rather low (21%), the precision is quite high (92%). In fact, this precision value is considerably higher than the precision of the Naive Bayes classifier at the corresponding level of recall. This result indicates the value of representing grammatical structure when learning information extractors. We believe that the recall level of our relational learner can be improved by tuning the set of background relations it employs, and we are investigating this issue in our current research.

Related Work

Several other research groups have addressed the task of information extraction from biomedical texts. Our research differs considerably, however, in the type of knowledge we are trying to extract and in our approach to the problem.

A number of groups have developed systems for extracting *keywords* from text sources. Andrade and Valencia (1997) describe a method for extracting keywords characterizing functional characteristics of protein families. This approach identifies words that are used much more frequently in the literature for a given family than in the literature associated with other families. In similar work, Ohta *et al.* (1997) extract keywords using an information-theoretic measure to identify those words that carry the most information about a given document. Weeber and Vos (1998) have developed a system for extracting information about adverse drug reactions from medical abstracts. Their system isolates words that occur near the phrase "side effect" and then uses statistical techniques to identify words that possibly describe adverse drug reactions. In all of these research efforts, the information-extraction task is to identify and extract informative words related to some topic. In our work, on the other hand, we are focusing on extracting instances of specific target relations.

Fukuda *et al.* (1998) consider the task of recognizing protein names in biological articles. Their system uses both orthographic and part-of-speech features to recognize and extract protein names. Whereas the task we are addressing is to extract *relation* instances, Fukuda *et al.* are concerned with extracting instances of a *class*, namely proteins.

The prior research most similar to ours is that of Leek (1997). His work investigated using hidden Markov models (HMMs) to extract facts from text fields in the OMIM (On-Line Mendelian Inheritance in Man) database. The task addressed by Leek, like our task, involved extracting instances of a binary relation pertaining to location. His location relation, however, referred to the positions of genes on chromosomes. The principal difference between Leek's approach and our approach is that his HMMs involved a fair amount of domain-specific human engineering.

Discussion and Conclusions

One may ask whether the learned classifiers we described in this paper are accurate enough to be of use. We argue that, for many tasks, they are. As discussed in the Introduction, two of the motivating applications for our work are (i) providing structured summaries of particular biological objects, and (ii) supporting discovery by eliciting connections among biological objects. As demonstrated by the work of Swanson *et al.* (Swanson & Smalheiser 1997), even word co-occurrence predictors can be quite useful for these tasks. Therefore, any method that can provide a boost in predictive power over these baselines is of practical value. For tasks such as automatic genome annotation, where the predictions made by the information extractors would be put directly into a database, the standard for accuracy is higher. For this type of task, we believe that extraction routines like those described in this paper can be of value either by (i) making only high-confidence predictions, thereby sacrificing recall for precision, or (ii) operating in a semi-automated mode in which a person reviews (some) of the predictions made by the information extractors.

Perhaps the most significant contribution of our work is the approach to using "weakly" labeled training data. Most previous work in learning information extractors has relied on training examples consisting of documents precisely marked with the facts that should be extracted along with their locations within the document. Our approach involves (i) identifying existing databases that contain instances of the target relation, (ii) associating these instances with documents so that they may be used as training data, and (iii) dividing the documents into training instances and weakly labeling these instances (e.g. by assuming that all sentences that mention a protein and a subcellular location represent instances of the subcellular-localization relation) We believe that this approach has great promise because it vastly reduces the time and effort involved in assembling training sets for inducing information extractors. Currently, we are investigating modifying the learning process to take into account the nature of weakly labeled training data. Specifically we are developing objective functions that are biased towards covering at least one sentence per positive instance instead of equally weighting all sentences labeled as positive.

We have numerous other plans to extend the work presented here. First, we are currently using our learned information extractors to help populate a protein-localization knowledge base being developed at Carnegie Mellon University. Second, we plan to learn information-extraction routines for all of the relations mentioned in the Introduction. Third, we plan to investigate ways in which existing sources of domain knowledge, such as the Unified Medical Language System (National Library of Medicine 1999b), can be leveraged to learn more accurate extraction routines. Fourth, we plan to address the task of extracting instances that are not represented by individual sentences. Fifth, we plan to extend our relation-extraction methods so that they can take into account factors that may qualify a fact, such as its temporal or spatial scope.

In summary, we believe that the work presented herein represents a significant step toward making textual sources of biological knowledge as accessible and interoperable as structured databases.

References

Andrade, M. A., and Valencia, A. 1997. Automatic annotation for biological sequences by extraction of keywords from MEDLINE abstracts. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, 25–32. Halkidiki, Greece: AAAI Press.

Boland, M. V.; Markey, M. K.; and Murphy, R. F. 1996. Automated classification of protein localization patterns. *Molecular Biology of the Cell* 8(346a).

Califf, M. E. 1998. *Relational Learning Techniques for Natural Language Extraction.* Ph.D. Dissertation, Computer Science Department, University of Texas, Austin, TX. AI Technical Report 98-276.

Cardie, C. 1997. Empirical methods in information extraction. *AI Magazine* 18(4):65–80.

Cestnik, B. 1990. Estimating probabilities: A crucial task in machine learning. In *Proceedings of the Ninth European Conference on Artificial Intelligence*, 147–150. Stockholm, Sweden: Pitman.

Cowie, J., and Lehnert, W. 1996. Information extraction. *Communications of the ACM* 39(1):80–91.

Domingos, P., and Pazzani, M. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29:103–130.

Freitag, D. 1998. Multistrategy learning for information extraction. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 161–169. Madison, WI: Morgan Kaufmann.

Fukuda, K.; Tsunoda, T.; Tamura, A.; and Takagi, T. 1998. Toward information extraction: Identifying protein names from biological papers. In *Pacific Symposium on Biocomputing*, 707–718.

Genome Annotation Consortium. 1999. The genome channel. http://compbio.ornl.gov/tools/channel/.

Hodges, P. E.; Payne, W. E.; and Garrels, J. I. 1998. Yeast protein database (YPD): A database for the complete proteome of saccharomyces cerevisiae. *Nucleic Acids Research* 26:68–72.

Karp, P.; Riley, M.; Paley, S.; and Pellegrini-Toole, A. 1997. EcoCyc: Electronic encyclopedia of E. coli genes and metabolism. *Nucleic Acids Research* 25(1).

Lathrop, R. H.; Steffen, N. R.; Raphael, M. P.; Deeds-Rubin, S.; Pazzani, M. J.; Cimoch, P.; See, D. M.; and Tilles, J. G. 1998. Knowledge-based avoidance of drug-resistant HIV mutants. In *Proceedings of the Tenth Conference on Innovative Applications of Artificial Intelligence.* Madison, WI: AAAI Press.

Leek, T. 1997. Information extraction using hidden markov models. Master's thesis, Department of Computer Science and Engineering, University of California, San Diego, CA.

Lewis, D. D., and Ringuette, M. 1994. A comparison of two learning algorithms for text categorization. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, 81–93.

Mitchell, T. M. 1997. *Machine Learning.* New York: McGraw-Hill.

National Center for Biotechnology Information. 1999. Entrez. http://www.ncbi.nlm.nih.gov/Entrez/.

National Library of Medicine. 1999a. Pubmed. http://www.ncbi.nlm.nih.gov/PubMed/.

National Library of Medicine. 1999b. Unified medical language system. http://www.nlm.nih.gov/research/umls/umlsmain.html.

Ohta, Y.; Yamamoto, Y.; Okazaki, T.; Uchiyama, I.; and Takagi, T. 1997. Automatic construction of knowledge base from biological papers. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, 218–225. Halkidiki, Greece: AAAI Press.

Pearl, J. 1988. *Probabalistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA: Morgan Kaufmann.

Porter, M. F. 1980. An algorithm for suffix stripping. *Program* 14(3):127–130.

Provost, F., and Fawcett, T. 1998. Robust classification systems for imprecise environments. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 706–713. Madison, WI: AAAI Press.

Quinlan, J. R. 1990. Learning logical definitions from relations. *Machine Learning* 5:239–2666.

Richards, B. L., and Mooney, R. J. 1992. Learning relations by pathfinding. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 50–55. San Jose, CA: AAAI/MIT Press.

Riloff, E. 1996. An empirical study of automated dictionary construction for information extraction in three domains. *Artificial Intelligence* 85:101–134.

Riloff, E. 1998. The sundance sentence analyzer. http://www.cs.utah.edu/projects/nlp/.

Rost, B. 1996. PHD: Predicting one-dimensional protein structure by profile based neural networks. *Methods in Enzymology* 266:525–539.

Slattery, S., and Craven, M. 1998. Combining statistical and relational methods for learning in hypertext domains. In *Proceedings of the Eighth International Conference on Inductive Logic Programming.* Springer Verlag.

Soderland, S. 1996. *Learning Text Analysis Rules for Domain-speific Natural Language Processing.* Ph.D. Dissertation, University of Massachusetts. Department of Computer Science Technical Report 96-087.

Soderland, S. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning.*

Swanson, D. R., and Smalheiser, N. R. 1997. An interactive system for finding complementary literatures: a stimulus to scientific discovery. *Artificial Intelligence* 91:183–203.

Weeber, M., and Vos, R. 1998. Extracting expert medical knowledge from texts. In *Working Notes of the Intelligent Data Analysis in Medicine and Pharmacology Workshop*, 23–28.

Xu, Y.; Mural, R. J.; Einstein, J. R.; Shah, M. B.; and Uberbacher, E. C. 1996. GRAIL: A multi-agent neural network system for gene identification. *Proceedings of the IEEE* 84(10):1544–1552.