BIOINFORMATICS

Vol. 1 no. 1 2001
Pages 1–10

Disambiguating Proteins, Genes, and RNA in Text: A Machine Learning Approach

Vasileios Hatzivassiloglou[1], *Pablo A. Duboué*[1] *and Andrey Rzhetsky*[2]

[1]*Department of Computer Science, Columbia University, 1214 Amsterdam Avenue, New York, NY, 10027, USA and* [2]*Genome Center and Department of Medical Informatics, Columbia University, 1150 St. Nicholas Avenue, New York, NY, 10032, USA*

ABSTRACT

We present an automated system for assigning protein, gene, or mRNA class labels to biological terms in free text. Three machine learning algorithms and several extended ways for defining contextual features for disambiguation are examined, and a fully unsupervised manner for obtaining training examples is proposed. We train and evaluate our system over a collection of 9 million words of molecular biology journal articles, obtaining accuracy rates up to 85%.

Contact: vh@cs.columbia.edu

INTRODUCTION

Discovery of new genes and new gene functions has become a daily routine as opposed to the painfully slow progress of the recent past; an unprecedented number of scientists both in academia and in industry collaborate in colossal genome sequencing and functional analysis projects. But precisely the same features that make this situation exciting create a new problem: Individual researchers are often unable to keep up with the rate of information accumulation. In many fields of molecular biology, and particularly in signal transduction, thousands of new results are published each year, and relationships between the information in different articles is often not immediately apparent, even if a researcher manages to read all relevant published articles.

A promising approach to handle the resulting information overload is to automate the process of knowledge extraction, using data mining techniques on the text of published articles, extracting novel information and relationships between biological entities such as genes, proteins, and RNA, and encoding that information in a computer-accessible form, so that further reasoning and analysis can be performed on it. At Columbia University, we are pursuing the design and development of a fully automatic system, GeneWays, for extracting knowledge of this type. GeneWays is a collaborative effort involving four departments (Computer Science, Medical Informatics, and Genome Center at Columbia University and Computer Science at Queens College CUNY) and aims at the extraction of molecular pathways from on-line research articles with natural language processing (NLP) techniques. The general architecture of the language-processing part of our system includes seven major modules: The collection module retrieves newly published articles from the Internet on a daily basis; the preprocessing module transforms the HTML encoding to a more rigidly specified (and thus understandable to the machine) XML representation, and enriches it with various types of additional language information (such as part-of-speech tags and sentence breaks); the term extraction module identifies words and phrases (*terms*) that are content-bearing and of interest to biological applications (such as genes, proteins, and small molecules); and the disambiguation module assigns a class label (such as gene or protein) to each term occurrence. This information is used by the fifth module to propose general patterns and constraints involving biological terms, and, together with a supplemental lexicon and a grammar, by the sixth module which extracts relationships between terms from a single article; these tentative relationships are validated across articles and against existing knowledge by the seventh module, and, if they pass muster, are entered in a persistent database. Modules one to four are currently fully implemented, substantial progress has been made on module six, and modules five and seven are the topic of future work.

This paper focuses on the fourth module in the above architecture, which disambiguates words or phrases known to be terms in this domain by assigning specific class labels to them. This disambiguation task is an important milestone for allowing us to reason with classes (e.g., proteins) rather than specific phrases. For example, it allows us to automatically discover, in a subsequent module, that *proteins activate genes*, and not the other way around. This is done by collecting information on patterns of the form "X activates Y" and observing that the X's tend to be proteins, while the Y's genes. Even the fact that "activate" is an important verb in the biology domain is automatically discovered, by observing that it co-occurs a lot with proteins and genes. Although our current implementation of the sixth module uses mostly manually developed rules capturing these constraints (Friedman et al., 2001), we are moving towards automating the task of learning these patterns from the texts. In contrast with other current work on finding relationships from biological texts (Blaschke et al., 1999; Proux et al., 2000), we are aiming to extract both general patterns and their instantiations in the text rather than matching prespecified patterns to the articles.

The kind of sense disambiguation addressed here is essential for processing text in any domain where, for reasons of terminological economy, terms are associated with multiple meanings. The biology domain offers a prime example of this multiplicity of meanings, since every protein has an associated gene with often the same name. Further, genes and their transcripts (mRNA, rRNA, tRNA and the like) often share the same name as well. Often, an article will refer to the protein, gene, and RNA senses of a term in close proximity, relying on the reader's expertise and the surrounding context for disambiguation. For example, *SBP2* is listed as a gene/protein in the GenBank (Benson et al., 1999) database. In one of our source articles (Copeland et al., 2000) we find the following sentences:

- "By UV cross-linking and immunoprecipitation, we show that **SBP2** specifically binds selenoprotein mRNAs both in vitro and in vivo."
- "The **SBP2** clone used in this study generates a 3173 nt transcript (2541 nt of coding sequence plus a 632 nt 3' UTR truncated at the polyadenylation site)."

In the first sentence the highlighted occurrence of SBP2 is a protein, while the highlighted occurrence in the second sentence is a gene.

The ambiguity problem often becomes an issue for human readers, as evidenced by the occasional inclusion of disambiguating information (e.g., "the SBP2 gene") by the authors of an article, and by the establishment of typographic conventions involving capitalization or italicization by some journals (e.g., *Gene*). However, not all online articles strictly follow these typographic conventions, and italics and capitalization are used for other text purposes as well (e.g., emphasis, or for sentence-initial words and proper nouns). Consequently, the disambiguation task would be hopeless for an automated system without a powerful technique for resolving ambiguity. We present such a method in this work, drawing from approaches used in statistical natural language processing.

Our method uses the context of known occurrences of genes, proteins, and mRNA to learn weights for elements in that context, so that applying these weights to the elements in the context of an unknown occurrence would result in the accurate classification of that occurrence as gene or protein.[†] We innovate by using as learning features not just the words around the term but also positional and morphological information. Since obtaining labeled genes and proteins for the learning phase requires laborious efforts by domain experts who would annotate the articles, we identify instead specific types of context where our system can confidently classify a term as gene, protein, or mRNA and use these as the input to our learning algorithm, generalizing to all types of occurrences of biological terms. This makes our approach an example of *unsupervised learning*, where only the raw text and no human input or annotation is available to the system.

In the next section, we situate our approach relative to other efforts for knowledge extraction in medical informatics, biology, and natural language processing. We then describe our process for data collection and preprocessing, as well as a particular way for extracting terms, which we use in the experiments reported in this paper. Then we present different ways of defining features for modeling the context of each term occurrence and three learning algorithms that we applied to our data. We follow with results and evaluation scores on a 9 million word collection of articles, and compare system- and human-assigned classes on a smaller set of terms. These results indicate that the system produces the correct label 78–84% of the time, making it a valuable tool for further statistical and knowledge-based analysis of the texts.

UNSUPERVISED LEARNING VERSUS ALTERNATIVES

Natural language systems have been successfully developed for various tasks involving text processing. However, in a practical application, the need for high specificity (low rate of false positives) in the output that the system extracts from the text often leads the system's designers to follow a knowledge intensive approach: Detailed models of human language processing, often in the form of a *semantic grammar* (Baud et al., 1992) (a grammar with both syntactic and classification information in its non-terminals), is painstakingly encoded by hand. The resulting system can then perform its task with an acceptably low error rate. However, this approach is applicable only to fairly narrow domains, where syntactic and semantic knowledge can be modeled with a reasonable amount of human effort.

The medical informatics domain is perhaps the best example of deployed natural language systems, which almost always follow the above knowledge-based approach, since specificity is of paramount importance in medical

[†] A separate stage filters and classifies other types of biological entities from non-ambiguous classes such as small molecules, using table lookup.

applications. MedLEE is an example of such a system; using a domain specific lexicon and semantic grammar, it achieves performance comparable to that of human experts in processing mammography reports (Hripcsak et al., 1995). Knowledge-intensive approaches involving a grammar have also been applied to the task of extracting information from biology texts (Park et al., 2001; Yakushiji et al., 2001).

An alternative that sacrifices some of the accuracy offered by the methods that model human knowledge, but obtains much broader generality and adaptability, is to rely instead on data-intensive, learning methods guided by a much weaker model of language behavior. Empirical methods of this type have gradually gained prominence in the research side of natural language processing during the 1990s. Since creating in advance a full lexicon of genes and proteins and rules for disambiguating each of them in context is impractical for more than a few thousand terms, and does not allow the easy addition of information on newly discovered genes, we opted for using a statistical method for the disambiguation stage of the system. Subsequent modules can then use this information to apply explicit syntactic knowledge models to classes covering thousands of individual terms, and we in fact use an adapted version of MedLEE (Friedman et al., 2001) for part of the relationship extraction module.

The task of assigning gene or protein labels to terms is a case of *word sense disambiguation* (albeit a hard one, because of the pervasive ambiguity in the biology domain). Earlier work on this problem with general language terms (e.g., disambiguating between the use of the word *bank* as a financial institution and as the slope next to a river) proposed modeling the context of each ambiguous word as a vector of neighboring words (Brown et al., 1991; Gale et al., 1992); for example, words such as “teller” and “money” would likely indicate that a nearby ambiguous *bank* referred to the financial institution sense. These methods obtain classification accuracies (percentage of correctly disambiguated cases) of 65-92% depending on the word being disambiguated and the alternative senses (choosing between *bank/institution* and *bank/river* is much easier than choosing between *bank/institution* and *bank/building* (Buitelaar, 1998)).

A drawback to these statistical learning techniques as originally formulated is their requirement for labeled training examples, which usually can only be obtained by manual annotation. Several techniques have been proposed to address this practical deficiency, including bootstrapping (Hearst, 1991), using parallel texts in different languages (Dagan and Itai, 1994), constructing pseudo-words for training (Gale et al., 1992), and using contextual evidence that indicates that some of the unlabeled terms belong to a particular class (Yarowsky, 1995). We apply this last approach to the biology domain, by noting that 9,187 (2.65%) of the occurrences of genes, proteins, and mRNAs in our text collection are readily disambiguated by the authors of the article, who include the word “gene”, “protein”, or “mRNA” right after the ambiguous word. Although these cases (which we term *non-ambiguous occurrences*) are probably the hardest for humans to disambiguate without the explicit “gene”, “protein”, or “mRNA” following them, it is plausible to assume that some of the other words near a non-ambiguous occurrence are also indicative of its status as gene, protein, or mRNA. Using only this information, we trade some of the increased accuracy that could be obtained with hand-labeled data for a much greater set of unlabeled (but disambiguated in the manner described above) examples, making our method unsupervised.

DATA COLLECTION

An automated script is used to download articles that appear in HTML (HyperText Markup Language) format on the Internet, at prespecified journal publishers’ web sites or via keyword searches through the PubMed database (http://ncbi.nlm.nih.gov). Downloaded articles often contain extraneous information (such as PubMed headers, links to facilitate site navigation, and information about the journal); we have written scripts to transform the HTML representation to a more controlled and structured version in XML (eXtended Markup Language) that contains only content-bearing tags. A DTD (Document Type Definition) that defines our target document representation has been specified, and we achieve the translation from HTML to the XML representation using the publicly available HTML::Parse and LT XML extraction and processing tools (Brew et al., 2000). Converting the articles to XML rather than working directly with the HTML source allows us to standardize the format of our text data across different journals, to strip HTML tags that carry no useful information for purposes of text analysis, and to enrich the document with additional tags.

The added XML tags mark word, sentence, paragraph, and section boundaries, non-textual material such as figures and tables, and specific kinds of text content that stand out from the main part of the document (such as author information and references). We use formatting information in the original HTML document to detect section and paragraph breaks, the MXTerminator statistical sentence detector (Reynar and Ratnaparkhi, 1997) retrained on biological text for finding sentence boundaries, and a word tokenizer we have implemented as a collection of pattern-matching finite state automata for identifying individual words, punctuation, and symbols in the text. We also add part-of-speech information to each token in the input text, using a statistical part-of-speech tagger (Brill, 1992). Figure 1 shows a portion of a sample

```
<!DOCTYPE PAPER SYSTEM "/proj/nlp/genome/xml/dtd/paper.dtd" >
<PAPER>
...
<AUTHORS> <AUTHOR>Vassilios Alexiadis</AUTHOR>
...
<AUTHOR>Claudia Gruss</AUTHOR>
</AUTHORS>
<JOURNAL>The EMBO Journal Vol. 17,pp. 3428-3438, 1998</JOURNAL>
<ABSTRACT>
...
</ABSTRACT>
<BODY>
<DIV DEPTH="1">
<HEADER>Introduction</HEADER>
  <P><S>
    <W C="W" P="NNP">Little</W> <W C="W" P="VBZ">is</W>
    <W C="W" P="VBN">known</W> <W C="W" P="IN">about</W>
    <W C="W" P="WRB">how</W> <W C="W" P="DT">the</W>
    <W C="W" P="NN">replication</W> <W C="W" P="NN">machinery</W>
    ...
  </S></P>
</DIV></BODY></PAPER>
```

Fig. 1. Sample portion of annotated document.

article, annotated with the XML tags described above.

Identifying terms in the text is a research topic on its own, and several approaches can be used. For the experiments reported here, we employ a simple lookup method over the GenBank database (Benson et al., 1999), which, as of February 2001, contains 204,177 gene/protein/RNA names. Gene or protein names can be phrases composed of common English words, either with a transparent meaning (e.g., *antifreeze protein precursor*) or with more obscure origins (e.g., *forever young* in the plant *Arabidopsis thaliana*); most, however, are strings of letters and digits not occurring in regular English text, such as *HI0509* or *SBP2*. Since our goal here is not term extraction but rather term disambiguation, we only consider as terms those entries in GenBank that either consist of multiple words or, if single words, do not appear in the 80,531-entry lexicon of common English words used by Brill's statistical part-of-speech tagger (Brill, 1992). This filtering step eliminates only 0.9% of the valid gene and protein names from consideration and significantly increases the accuracy of the recognized terms. It is possible to replace this simple term extractor with a more sophisticated method, relying on frequency and distributional information (Justeson and Katz, 1995) and complementing that with techniques that utilize approximate string matching to identify term variants and new terms similar to old ones, such as the method proposed by Krauthammer et al. (2000) specifically for biological terms.

Since terms do not neatly correspond to single words, our term identification method also looks for terms that appear as part of a longer word and for multiword terms. We break any word in the text at hyphens and allow for matches between one or more of these divisions and GenBank; this allows us to detect, for example, *gp41* as a term in *gp41-mediated*. We also match, at any given word, that word and the longest possible subsequence of words immediately preceding it against GenBank, allowing for multiword terms such as *mothers against decapentaplegic*.

With our document representation enhanced with term boundary information, we subsequently separate the terms in two categories: those immediately followed by a disambiguating word ("gene", "protein", or "mRNA"), and those that are not. The former are used during system training and for the automated evaluation, as described in the next two sections. Naturally, predictions are made for all terms, whether locally disambiguated or not.

Using the above approach, we collected 1,374 articles from the European Molecular Biology Organization (EMBO) journal, spanning the years 1997 to 2000. This represents 9,003,923 words of text, taking up 314 Mb (including markup). Our term extraction method identifies 346,519 terms (protein, gene, or mRNA names) in this collection, of which 9,187 (2.65%) are non-ambiguous occurrences.

LEARNING METHODOLOGY

Features

Words that appear near a term (whether a term for which the class label is known, during training, or one for which a label should be predicted) form the basic *features* to which contextual information is mapped to for machine learning (see references on sense disambiguation listed earlier). Our approach to feature definition extends earlier work by considering additional positional information in addition to the words themselves. For example, if the word "activates" immediately precedes a term, it is likely that the term is a gene, while protein is the most likely class if "activates" follows a term. We considered three ways of incorporating positional information into our features:

- Given a term, we collect all words near the term in a vector of counts representation (a *word bag*[‡] approach, with no positional information).
- Rather than using a single word bag for each term occurrence, we separate the nearby words into two bags: one for the words before the term, and another for the words after the term.
- Each word is annotated with its distance from the term, and we keep separate counts for each different position (so our features become, for example, $X/{+2}$ and $X/{-1}$ rather than just a word X).

We also consider several options for defining our features that utilize morphological, distributional, and shallow syntactic information to either further refine the features or to conflate several strings to the same feature:

- **Capitalization**. Word strings that differ only in case can be mapped to the same feature. Differences in case are often due to typographic conventions (e.g., sentence-initial words) rather than differences in content.
- **Part-of-speech**. Keeping a word's part of speech (e.g., noun or verb) allows for limited disambiguation of that word (for example, *structure/N* vs. *structure/V*). Different meanings may indicate different preferences for specific term classes.
- **Stopwords and similarly distributed words**. Common English words (such as articles ("the") and prepositions ("in")) make up a large percentage of the text, but often do not help in classification tasks since they are not strongly associated with any of the alternative classes. Extending this concept, we automatically identify during training other words, not in a standard stopword list, that are approximately equally distributed between the alternative classes. We select for removal those words for which a chi-square test contrasting their distributions across the alternative classes does not yield a statistically significant difference at the 5% level.
- **Stemming**. Stemming is an automated process that maps inflected forms of the same root to a common feature, so that, for example, *phosphorylate* and *phosphorylation* are treated as the same feature.

[‡] A bag is another term for a multiset, i.e., a set where multiple occurrences of the same element are allowed.

In all cases, we omit from the features the word "gene", "protein", or "mRNA" that always occurs right after the term (we use this information to label training and evaluation instances with their correct class). When the system is run in prediction mode, it automatically labels non-ambiguous occurrences using the immediately following disambiguating word; but this information is not used when we train or when we evaluate over non-ambiguous occurrences.

Defining what is a term's "nearby" words is an issue that has received considerable attention in the sense disambiguation literature, with research showing that sometimes disambiguating information can be found far from the term's occurrence (Gale et al., 1992). We define the neighborhood of a term as all words in a window extending N words to the left and N words to the right of the term. N is determined empirically from the training data; we explain this step later in this section.

The three learning techniques

We considered three established learning techniques for constructing a prediction model over the extended word features of the previous subsection. The first method is *naive Bayesian learning* (Duda and Hart, 1973). The method aims to assign to a term occurrence the class c that maximizes $P(c|\mathcal{E})$, where $\mathcal{E}$ is the evidence available to the machine learning algorithm for that occurrence (i.e., the features $f_1, f_2, \ldots, f_k$ in the term's context). Using Bayes' rule, $P(c|\mathcal{E})$ can be written as

$$P(\mathcal{E}|c) \times \frac{P(c)}{P(\mathcal{E})} = P(f_1, f_2, \ldots, f_k|c) \times \frac{P(c)}{P(f_1, f_2, \ldots, f_k)} \tag{1}$$

where $P(c)$ is the a priori probability of class c. The a priori probability of the evidence $P(f_1, f_2, \ldots, f_k)$ is the same for all classes c, and hence does not affect the maximization of (1). So we equivalently maximize

$$P(c) \times P(f_1, f_2, \ldots, f_k|c) \tag{2}$$

Estimating $P(f_1, f_2, \ldots, f_k|c)$ for a large k is however impossible for most applications due to the sparseness of available data. At this point, we make an independence assumption for the features f_i (hence the "naive" qualifier in the name of the method), and rewrite (2) as[§]

$$P(c) \times \prod_{i=1}^{k} P(f_i|c)$$

Although the assumption of independence between features is obviously a simplification (Mosteller and Wallace, 1984), the method frequently offers a good enough approximation to the true classification odds among the classes. For example, *terminator*, *genomic*, *replacing*, *translocations*, and *allelic* appear among the top 25 (out of 25,000) discriminators for the gene class.

The second method we considered is *decision trees*, in particular the C4.5 implementation of decision tree learning (Quinlan, 1993). Decision trees recursively partition the feature space into areas corresponding to each class label. First, the feature that discriminates most between the classes is selected for the root of the tree, and, depending on the presence or absence of that feature, the algorithm proceeds to the left or right subtree. Subtrees are recursively constructed in a similar manner, until eventually the process terminates at a leaf, associated with one of the class labels.

A third method conceptually similar to decision trees is *inductive rule learning*; we experimented with the RIPPER implementation of this approach (Cohen, 1996). In this method, rules involving tests on features are iteratively constructed; these rules map a particular combination of features to a class label, and are applied sequentially during prediction, so that rules where the system has the highest confidence are applied first. Note that it is possible to convert decision trees to a similar rule format (by tracing the path followed from the root to a leaf), so both of these approaches offer the advantage of allowing a human observer to realize what makes the classifier work. Negative information (i.e., that a feature *does not* appear near a term occurrence) is made explicit in the rules, although it is implicitly used by all three methods. Figure 2 shows a sample of the rules automatically learned by C4.5; for example, Rule 530 captures the fact that genes *encode* information while Rule 408 captures the fact that references to genes are often followed by a citation to the work of other researchers (the ET comes from *et al.*). Rule 408 is in fact a good example of evidence that can be automatically detected from text regularities but would be unlikely to be included in a knowledge-based system.

[§]In the actual implementation of the Bayes method, all calculations are carried on a logarithmic scale, to preserve numerical accuracy with finite floating point representations.

```
Rule 408: after DEVELOPMENT is present
          after ET is present
          before DATA is NOT present
          ⟹ class gene [91.7%]

Rule 516: before THAT is NOT present
          before FRAGMENT is NOT present
          before ALLELE is present
          ⟹ class gene [93.9%]

Rule 530: after ENCODES is present
          before ENCODES is NOT present
          ⟹ class gene [96.5%]
```

Fig. 2. Sample rules produced by C4.5.

Experimental design

Systems that automatically learn rules or estimate parameters from a certain amount of training data always run the risk of adapting too much to pecularities of the data set used during training, which may not reflect general properties. To check against this potential pitfall, it is imperative to evaluate on data unseen by the system during training. However, the common approach of dividing the available data into separate training and test subsets does not make the most efficient use of the data. Another alternative is *n-fold cross-validation*, in which the total available data is split into n equal parts, and n training/test cycles are performed. During each of these cycles, one of the n parts is kept aside for testing, and the remaining $n - 1$ parts are used for training; the evaluation scores reported are the averages of the scores obtained during each cycle. In this way, all of the data is eventually used for both training and testing, but the data in each test set is never accessible to the system during training for that test set.

We perform our training and evaluation experiments using 10-fold cross-validation, with an additional refinement. We mentioned earlier that the size of the window of neighboring words N that defines the features related to each term occurrence is a parameter of the learning process. Obviously, large values of N have a smoothing effect, with the attendant advantages and disadvantages (capturing dependencies at longer range, but also dilluting the effect of the words closest to the term). To estimate the optimal value for N, we set aside 1/10th of our data and apply 10-fold cross-validation to it, running the learning algorithm successively for N from 2 to 35. The N that leads to the best overall classification accuracy during this phase is then fixed as the window size for the full training and evaluation phase, which is performed on the remaining 9/10th's of the data, again using 10-fold cross validation.

RESULTS AND EVALUATION

We have experimentally compared the performance of the disambiguation system under various learning conditions, using the different learning algorithms and the different options for constructing features described earlier. As our measure of performance we use the overall *rate of classification accuracy*, i.e., the total percentage of correct decisions made by the classification algorithm. This is always equal to 100% minus the error rate. Other measures of performance, such as precision/recall and specificity/sensitivity can be used to describe the system's performance on particular categories (e.g., for genes only) if an application assigns more importance to the accurate detection of terms from that particular category. Space does not permit listing these detailed scores for all experiments, but we report precision, recall, F_1-measure (van Rijsbergen, 1979), specificity, and sensitivity for the final experiment that uses the entire amount of data and the best combination of features.

We base our comparisons between algorithms and features on relative performance on locally disambiguated cases (non-ambiguous occurrences, i.e., those immediately followed by a disambiguating word), since by using automatically labeled cases we can evaluate close to 10,000 disambiguation decisions. However, since it is possible that the distributional characteristics of words in the context of non-ambiguous occurrences are different than those of ambiguous terms in general, we have also manually labeled a smaller set of 550 ambiguous occurrences, and we evaluate our final combination of features on that set as well.

We considered two target sets of labels for the terms: classifying them as genes, proteins, or messenger RNA (mRNA), or classifying them as genes or proteins only and ignoring occurrences of mRNAs in our training and test data sets. The former, three-way classification is in principle more useful, representing a more accurate model of term classes. But with a finite amount of data, and given the relatively low rate of occurrence of mRNAs in the text (13.93% of non-ambiguous terms), lower performance on the mRNA category may lower overall system accuracy. For applications that are primarily considering the distinction between genes and proteins, the two-way classification offers an alternative with higher expected accuracy.

Effects of the learning algorithm We first tested the three learning algorithms on about one third of our collection of articles. Their relative performance was similar in both the three-way and two-way classification settings, with C4.5 outperforming RIPPER by one percentage point on absolute terms in the two-way case and two percentage points in the three-way case, and naive Bayes being practically indistinguishable from C4.5 in terms of accuracy in both cases (Table 1). However, naive Bayes is significantly faster in both training and prediction than the other two methods, by about an order of magnitude over the slowest C4.5, and this speed gain is important as we handle large sets of data. Hence, we have chosen naive Bayes as the learning algorithm for the remainder of our experiments.

Table 1. Accuracy of different learning techniques (cross-validated on one-third of our data).

	Two-way classification	Three-way classification
RIPPER	74.59%	66.75%
C4.5	76.61%	67.81%
Naive Bayes	76.57%	67.62%

Effects of feature definitions We subsequently examined alternatives for our feature definitions, summarized below:

- **Positional information**. We tested whether words should be used as features directly, or whether information about their position relative to the term should be encoded as well. We found that using full positional information (counting words at different positions as different features, even if they shared the same string of characters) universally lowered accuracy by as much as six percentage points; keeping just the sign of the positional difference (i.e., whether words are before or after the term) slightly decreased accuracy (by 1-1.5% on average). We believe that this effect is likely due to the sparseness of data introduced when the same character string is mapped to different features, and we will explore in the future models that condition the use of positional information according to the frequency of a given word.

- **Capitalization**. Mapping all words (but not the terms themselves) to lower case did not alter performance. We include this mapping in our final system to slightly reduce the total number of features and thus enhance speed and lower memory requirements.

- **Part-of-speech**. Using part-of-speech information as a local disambiguator of the meaning of words generally helped the overall accuracy of the system, but only modestly (less than 1% on absolute terms on average). This is probably due to the technical nature of the domain (which offers less opportunity for ambiguity on non-terms than general English does). Consequently, we include this information in our final definition of features.

- **Similarly distributed words**. Eliminating words that do not differ significantly in their distribution between the two or three classes has a small negative effect on

	Precision	Recall/ Sensitivity	F-measure	Specificity
Two-way/Gene	81.83%	87.23%	84.44%	81.90%
Two-way/Protein	87.28%	81.90%	84.50%	87.24%
Three-way/Gene	78.83%	80.16%	79.48%	83.64%
Three-way/Protein	80.57%	77.73%	79.12%	85.94%
Three-way/mRNA	69.00%	72.89%	70.86%	94.70%

Table 2. Evaluation metrics for particular categories of terms.

	Two-way	Three-way
Vs. non-ambiguous terms in test set	84.48%	78.11%
Vs. majority human model	69.40%	63.58%
Vs. full agreement human model	70.68%	65.85%
Baseline (non-ambiguous terms)	43.19%	50.18%
Baseline (majority human model)	45.36%	51.12%
Baseline (full agreement human model)	51.71%	55.50%

Table 3. Overall evaluation scores for predicted class labels, against non-ambiguous and manually labeled terms.

performance (0.2–0.5% on absolute terms, depending on other options for defining features). As a result, we decided not to include this option in the final system, but we note that it can significantly reduce the total number of features that need to be considered during machine learning, from approximately 25,000 to about 5,000 (depending on other experimental parameters). Therefore it offers a promising approach if a five-fold gain in computational efficiency justifies a small loss of classification accuracy.

- **Stopwords**. Unlike the removal of content words, removing extremely common function words is useful for both increasing performance (0.5–1.5% increase in accuracy) and reducing the feature space.
- **Stemming**. Mapping related word forms to their common root increased classification accuracy by 0.4% on absolute terms on average, and therefore is included in our final system.

Evaluation of the final feature combination We evaluated the experimentally determined best combination of options for defining features on our full data collection using 10-fold cross-validation. Table 2 shows detailed evaluation scores for particular classes of terms, while accuracy measurements for our overall results are shown in the first row of Table 3, indicating that the system achieves approximately 80% accuracy in the three-way classification and 85% accuracy in the two-way classification.

The second and third rows of Table 3 show system performance as compared to a set of manually labeled terms. These terms were chosen randomly among the majority of terms that are not locally disambiguated with an immediately following specifier. We randomly selected 15 articles from our collection, extracted 75 paragraphs at random from them, and highlighted the 550 terms there according to our simple term finder (i.e., by matching with GenBank and removing common English words). We then asked three experts (all holding Ph.D. or M.S. degrees in molecular biology, bioengineering, or mathematical genetics) to manually tag each term as gene, protein, mRNA, ambiguous, or wrongly extracted. The three experts used the "ambiguous" label sparingly (only 24 times in the 1,650 total manually assigned labels, and never twice for the same term occurrence across different evaluators), but nevertheless exhibited considerable disagreement. Their average pairwise rate of agreement was 77.58%, which indicates that the disambiguation task is hard even for human experts. We mapped the human-assigned labels to two models for comparing with our system: First, we considered the term occurrences for which a majority of the three evaluators agreed on a label (518 occurrences, or 94.18% of the total), and, second, the occurrences for which all three evaluators assigned the same label (381 occurrences, or 69.27%) of the total. From each of those models we removed cases where the humans thought that the term extracted was not a gene, protein, or mRNA[¶], and scored our system's predictions against the remaining occurrences in the model. For the three-way classification task, the comparison is straightforward; for the two-way task, we also removed any occurrences of mRNAs from the model.

We also estimate baseline models, as lower bounds on system performance. We construct the baseline models by assuming that such a classifier always predicts the most frequent class, and estimate class frequencies from the automatically extracted non-ambiguous cases, from the cases in the human majority model, and from the cases in the human full-agreement model. These baseline models always predict "protein" as the class label; their accuracies are listed in the last three rows of Table 3, respectively.

¿From Table 3, we observe that performance is significantly lower against the human-labeled models when compared to our automatically extracted non-ambiguous cases. There are two possible explanations for this discrepancy: Either the human experts are correct, in which case the decline in performance can be partially attributed to the fact that many of the selected cases appear to be

[¶] These are errors made by our simple text extractor that matches strings against GenBank.

harder to classify (as evidenced by the relatively low agreement rate between humans), and partially to some inconsistencies in labeling cases that our system does not currently handle (for example, in follow-up interviews, one of the evaluators mentioned that he was consistently marking tRNA or snRNA occurrences as mRNA, as that was the closest among our labels). Or, the system is actually doing a more consistent job than the humans and is penalized for making correct decisions when they are wrong. This intriguing possibility is supported by the fact that our system does better when evaluated against the non-ambiguous cases (while not examining the disambiguating words) than the humans do against each other. We plan additional experiments where we will mask the word following non-ambiguous terms (providing a setting identical to how our system works) and compare their disambiguating performance on that data to our system's on the same data. Nevertheless, the system achieves performance which under the worst interpretation is 7–14 percentage points below human performance and improves accuracy by a relative 20–95% over the baselines.

CONCLUSION

We have explored three learning techniques and several ways for defining contextual features for the problem of automatically disambiguating biological terms such as genes, proteins, and mRNAs. Rather than relying on extensive human markup for training our system, we utilize instead textual information for disambiguating a small part of our data with high confidence and use this as seed training material for predicting labels on all ambiguous terms in our collection. We have performed a large-scale evaluation on automatically disambiguated data from a 9 million word corpus, and a smaller study using manual annotations. During system design, we optimized both accuracy and computational efficiency, an important constraint when working with large data sets.

Our system demonstrates accuracy within the range of statistical sense disambiguation applications; but more importantly, since it is used within a larger text mining system, perfect accuracy is not needed for obtaining high quality results further down our pipeline. Unlike a system that has to extract information from a specific text passage, we collect information from a large corpus, where repeated occurrences of the same relationship or pattern are expected; and this aggregation of information shields us against a reasonable percentage of local mistakes. We nevertheless outperform significantly baseline disambiguation systems, achieve performance near that of the human agreement rate when evaluated against humans, and score higher than that rate within our set of locally non-ambiguous cases. It is also possible to make predictions primarily or exclusively for cases where the system's confidence in the classification decision (as measured by the relative Bayesian odds across classes) is high; this will likely increase the overall accuracy. In the future, we plan to refine several aspects of our system, such as its positional information model, as well as use its results for predicting relationships between classes of biological terms.

ACKNOWLEDGMENTS

We thank Pavel Morozov for implementing the article downloading software, Wubin Weng for helping transcribe the experts' annotations, and Shawn M. Gomez, Tomohiro Koike, and Sila Konur for serving as expert evaluators. This publication was supported in part by a grant from the Columbia University Center for Advanced Technology funded by the New York State Science and Technology Foundation. Any opinions, findings, or recommendations are those of the authors and do not necessarily reflect the views of the NYSSTF.

REFERENCES

Baud, R., A. Rassinoux, and J. R. Scherrer (1992). Natural language processing and semantical representation of medical texts. *Meth. Inf. Med.* **31**(2), 117–125.

Benson, D. A., M. S. Boguski, D. J. Lipman, J. Ostell, B. F. Ouellete, B. A. Rapp, and D. L. Wheeler (1999). GenBank. *Nucl. Acids Res.* **27**(1), 12–17.

Blaschke, C., M. A. Andrade, C. Ouzounis, and A. Valencia (1999). Automatic extraction of biological information from scientific text: Protein-protein interactions. In *Proc. 7th ISMB*, pp. 60–67.

Brew, K., D. McKelvie, R. Tobin, H. Thompson, and A. Mikheev (2000). The XML library LT XML version 1.2. Available from http://ltg.ed.ac.uk/software/.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proc. 3rd ANLP*, Trento, Italy.

Brown, P. F., S. A. della Pietra, V. J. della Pietra, and R. L. Mercer (1991). Word-sense disambiguation using statistical methods. In *Proc. 29th ACL*, Berkeley, California, pp. 264–270.

Buitelaar, P. (1998). *CoreLex: Systematic Polysemy and Underspecification*. Ph. D. thesis, Brandeis University.

Cohen, W. (1996). Learning trees and rules with set-valued features. In *Proc. 14th AAAI*.

Copeland, P. R., J. E. Fletcher, B. A. Carlson, D. L. Hatfield, and D. M. Driscoll (2000). A novel RNA binding protein, SBP2, is required for the translation of mammalian selenoprotein mRNAs. *EMBO Journal* **19**, 306–314.

Dagan, I. and A. Itai (1994). Word sense disambiguation using a second language monolingual corpus. *Comput. Linguist.* **20**(4), 563–596.

Duda, R. O. and P. E. Hart (1973). *Pattern Classification and Scene Analysis*. New York: Wiley.

Friedman, C., P. Kra, M. Krauthammer, H. Yu, and A. Rzhetsky (2001). GENIES: A natural-language processing system for the extraction of molecular pathways from journal articles. In *Proc. 9th ISMB*, Copenhagen, Denmark.

Gale, W. A., K. W. Church, and D. Yarowsky (1992). Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proc. 30th ACL*, pp. 249–256.

Hearst, M. A. (1991). Noun homograph disambiguation using local context in large text corpora. In *Using Corpora*, U. of Waterloo.

Hripcsak, G., C. Friedman, P. O. Alderson, W. DuMouchel, S. B. Johnson, and P. D. Clayton (1995). Unlocking clinical data from narrative reports: A study of natural language processing. *Ann. Intern. Med.* **122**, 681–688.

Justeson, J. S. and S. M. Katz (1995). Technical terminology: Some linguistic properties and an algorithm for identification in text. *Nat. Lang. Eng.* **1**(1), 9–27.

Krauthammer, M., A. Rzhetsky, P. Morozov, and C. Friedman (2000). Using BLAST for identifying gene and protein names in journal articles. *Gene* **259**, 245–252.

Mosteller, F. and D. L. Wallace (1984). *Applied Bayesian and Classical Inference: The Case of The Federalist Papers.* New York: Springer-Verlag.

Park, J. C., H. S. Kim, and J. J. Kim (2001). Bidirectional incremental parsing for automatic pathway identification with combinatory categorial grammar. *Pacif. Symp. Biocomp.* **6**, 396–407.

Proux, D., F. Rechenmann, and L. Julliard (2000). A pragmatic information extraction strategy for gathering data on genetic interactions. In *Proc. 8th ISMB*, La Jolla, Ca., pp. 279–285.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann.

Reynar, J. C. and A. Ratnaparkhi (1997). A maximum entropy approach to identifying sentence boundaries. In *Proc. 5th ANLP*, Washington, D.C.

van Rijsbergen, C. J. (1979). *Information Retrieval* (2nd ed.). London: Butterworths.

Yakushiji, A., Y. Tateisi, Y. Miyao, and J. Tsujii (2001). Event extraction from biomedical papers using a full parser. *Pacif. Symp. Biocomp.* **6**, 408–419.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. 33rd ACL*, pp. 189–196.