A Large Scale, Corpus-Based Approach for Automatically Disambiguating Biomedical Abbreviations

HONG YU
University of Wisconsin-Milwaukee
WON KIM
National Center for Biotechnology Information
VASILEIOS HATZIVASSILOGLOU
University of Texas
and
JOHN WILBUR
National Center for Biotechnology Information

Abbreviations and acronyms are widely used in the biomedical literature and many of them represent important biomedical concepts. Because many abbreviations are ambiguous (e.g., *CAT* denotes both *chloramphenicol acetyl transferase* and *computed axial tomography*, depending on the context), recognizing the full form associated with each abbreviation is in most cases equivalent to identifying the meaning of the abbreviation. This, in turn, allows us to perform more accurate natural language processing, information extraction, and retrieval. In this study, we have developed supervised approaches to identifying the full forms of ambiguous abbreviations within the context they appear. We first automatically assigned multiple possible full forms for each abbreviation; we then treated the in-context full-form prediction for each specific abbreviation occurrence as a case of word-sense disambiguation. We generated automatically a dictionary of all possible full forms for each abbreviation. We applied supervised machine-learning algorithms for disambiguation. Because some of the links between abbreviations and their corresponding full forms are explicitly given in the text and can be recovered automatically, we can use these explicit links to automatically provide training data for disambiguating the abbreviations that are not linked to a full form within a text. We evaluated our methods on over 150 thousand abstracts and obtain for coverage and precision results of 82% and 92%, respectively, when performed as tenfold cross-validation,

This research was supported in part by the National Science Foundation Innovative Technology Research Grant EIA-0121687 and the National Institute of Health Grant R01 GM61372-01A2.

Authors' addresses: H. Yu, Department of Health Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI 53211; email: yuh9001@dbmi.columbia.edu; W. Kim, J. Wilbur, National Center for Biotechnology Information, Bethesda, MD 20894; V. Hatzivassiloglou, Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083.

and 79% and 80%, respectively, when evaluated against an external set of abstracts in which the abbreviations are not defined.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Algorithms, Languages

Additional Key Words and Phrases: Word-sense disambiguation, machine learning, data mining, information retrieval

1. INTRODUCTION

Abbreviations and acronyms are widely used in technical domains, particularly in the biomedical literature [Fauquet and Pringle 1999]. Many abbreviations represent important terms [Rimer and O'Connell 1998; Federiuk 1999]. For example, the names of many clinical diseases and procedures, and common entities such as genes and proteins, have widely used abbreviations. However, abbreviations are frequently ambiguous. For instance, the abbreviation "CAT" denotes *chloramphenicol acetyl transferase*, *computer-aided testing, computer-automated tomography*, *choline acetyltransferase*, and *computed axial tomography* [Rimer and O'Connell 1998] depending on the context. Since most technical terms, especially multiword terms, have unique meanings within their domains, if we were to correctly map an abbreviation to its intended full form, we would equivalently identify the meaning of the abbreviation. This, in turn, would allow us to perform more accurate natural language processing (NLP) for information extraction and retrieval from the literature.

We distinguish between two types of abbreviation occurrences. First, abbreviations may be disambiguated (defined) near their occurrence in the text. This typically is achieved by linking the abbreviation and the intended full form with a linguistic construction such as a parenthetical expression, apposition, or the use of terms such as "i.e.," "that is," and equivalent expressions. The second type of abbreviation appears without the intended full form nearby. This type is both more prevalent and harder to disambiguate. In our earlier study, we analyzed ten randomly selected biomedical full-text articles and found that 75% of the total 358 abbreviations in these articles had never been defined [Yu et al. 2002]. In the following, we will show two abstracts to illustrate the importance of capturing the full forms of both types of abbreviation for the purpose of information retrieval:

> *A1.* The activity of inosine triphosphate pyrophosphohydrolase (ITPH) in human erythrocytes was found to be 1.50 ± 0.39 mumol of inosine triphosphate (ITP) hydrolysed x min-1 per g Hb, and no measurable amount of ITP was detected. When dipyridamole was added to the medium composed of adenosine, pyruvate and inorganic phosphate, ITPH activity was 1.18 ± 0.41, and at the same time ITP accumulation was 0.61 ± 0.31 mumol/g Hb. The negative correlation between ITPH activity and accumulation of ITP was $r = -0.87$ at P less than 0.001 [Kopff et al. 1990].

> *A2.* ITP in hemophiliacs may produce severe bleeding complications. We here report on an eight-year-old boy suffering from severe hemophilia A, who developed ITP and an acquired impaired immune function similar to AIDS. Steroid therapy reverted the thrombocyte count to normal, however it had to be discontinued because of a severe Cushing syndrome. The thrombocytopenia also responded to IgG-therapy and the patient is treated with a long term schedule according to Imbach. It is of interest that the impaired T-helper/T-suppressor cell ratio (0.45) improved to a value of 1.0 after initiation of this therapeutic regimen. We conclude from our observation that i.v. immunoglobulin therapy is of particular value for the treatment of ITP in patients with impaired cellular immunity [Zeitlhuber et al. 1984].

In the first abstract (A1), "ITP" is defined as *inosine triphosphate,* and subsequently used in that sense throughout the remainder of abstract. In the second, "ITP" is not defined within the abstract. When an information retrieval system searches for the keyword "ITP" both abstracts are retrieved, though "ITP" in the latter abstract denotes *idiopathic thrombocytopenic purpura.* Conversely, if we search directly for the phrase *idiopathic thrombocytopenic purpura,* A2 is not retrieved.

In this article, we build upon our earlier work which identifies with both high precision and recall defined abbreviations (i.e., cases where the abbreviations and intended full forms are explicitly linked) [Yu et al. 2002]. Using these simpler cases as training data, we postulate that features occurring near abbreviations of this kind will have a similar distribution to corresponding features for the same abbreviation *when it is used in the same sense*, that is, to represent the same (hidden, in this case) full form. This assumption leads us to apply supervised learning methods to what is essentially a classification problem, since both defined abbreviations and their full forms can be extracted automatically.

Our method does not depend upon a knowledge source for sense listing, although, as we discuss in Section 4, it can take advantage of existing semantic knowledge sources to extend the available feature set. We extract senses automatically from biomedical literature. Since we are able to obtain the data for training automatically, the system resembles an unsupervised approach in regard to the economy of effort required to accomplish the task. Although in this study we focus on the biomedical domain, as it provides a significant amount of text data for experimentation in addition to supplementary knowledge sources that can be used by automated systems, the methods described in this article apply to other technical domains, as well.

2. RELATED WORK

A number of systems have been developed to map abbreviations to intended full forms when the latter are explicitly defined in the article. Examples include Chang et al. [2006], Bowden et al. [1998], Hisamitsu and Niwa [1998], Yeates et al. [2000], Yoshida et al. [2000], Park et al. [2001], Pustejovsky et al. [2001], Adar [2002], Schwartz and Hearst [2002], and Yu et al. [2002] recall and precision ranged from 73% and 84% and [Bowden et al. 1998], respectively to 96% and 98% [Fukuda et al. 1998]. Note that system-wide performance comparison is difficult, due to a lack of common standards.

We had earlier developed AbbRE [Yu et al. 2002], which was specifically developed for use within the biomedical domain to map abbreviations to full forms when the latter are explicitly defined in biomedical articles. AbbRE operates through a set of manually annotated rules that assign matches between letters in abbreviations and words in the full forms. These rules include:

—*the first letter of an abbreviation must match the first letter of the meaningful word of the full form*;

—*the abbreviation must match the first letter of each word in the full form*;

—*the abbreviation letter must match consecutive letters of a word in the full form; and*

—*the abbreviation letter must match a middle letter of a word in the full form if the first letter of the word matches the abbreviation*.

AbbRE was evaluated in full-text biomedical articles and found to have 70% recall and 95% precision rates.

Fewer systems, however, have been developed for full-form recognition of abbreviations for which the intended full forms are not explicitly defined. Pustejovsky et al. [2001] applied a vector-space model, a typical information-retrieval technique, for acronym sense disambiguation. Given an article in which the abbreviation is ambiguous, the task is to use the article to retrieve a similar article in which the same abbreviation has a defined full form; this full form is then predicted to be the intended full form of the abbreviation to be disambiguated. Yet Pustejovsky et al. [2001] evaluated their method using only one abbreviation with four distinct meanings that appears in a small set of testing abstracts (merely 42 abstracts).

Liu et al. [2001] and Pakhomov [2002] applied word-sense disambiguation methods for disambiguating abbreviations in medical notes. Both methods obtained the senses of abbreviations from the unified medical language system (UMLS), a biomedical knowledge source that incorporates a small set of abbreviations and their possible full forms [Humphreys and Lindberg 1993]. Both exploited surrounding words as features for disambiguation. Liu et al. [2001], applied machine-learning approaches, including naïve Bayes and decision lists, and reported 92–97% precision on disambiguating 12 medical abbreviations. Pakhomov [2002] used statistical maximum-entropy techniques and reported 89% accuracy on disambiguating six medical abbreviations.

In this study, we present a word-sense disambiguation model for identifying the full forms of biomedical abbreviations in MEDLINE abstracts. Unlike previous models [Liu et al. 2001; Pakhomov 2002], we do not depend on an existing knowledge source for sense listing. Instead, we generated a sense dictionary automatically from the MEDLINE corpus. We also explored different learning features to aid the process of disambiguation. Our learning features include medical subject heading (MeSH) terms, which is a controlled vocabulary maintained by the National Library of Medicine (NLM). MeSH terms are manually assigned to each MEDLINE record by the NLM annotators for indexing purposes. The addition of MeSH terms presents a case of semantic class-based smoothing (the details of MeSH terms and smoothing will be described in Section 4).

We also propose a *full-form normalization* method to normalize variations among full forms. These variations are abundant in the biomedical literature. For example, one sense of "ITP" includes the varied full forms *immune allergic thrombocytopenia, immune mediate thrombocytopenia, immune thrombocytopenia*, and *immunothrombocytopenia*. It is important to recognize these variations so that abstracts incorporating the same sense of a given abbreviation can be clustered into one training group. We developed an algorithm (the details of which are given in Section 7, Step 3) that identifies the variations among full forms. We found that such normalization significantly enhances the performance of sense disambiguation (explained further in Section 14). Neither Liu et al. [2001] nor Pakhomov [2002] mentioned the full-form variation problem. Finally, we evaluated a large number of biomedical abbreviations (i.e., 60 abbreviations) over a large number of MEDLINE abstracts (over 150,000), whereas previous evaluations focus mainly on the medical domain with only a few abbreviations.

3. IDENTIFYING THE FULL FORMS OF UNDEFINED ABBREVIATIONS IS A CASE OF WORD-SENSE DISAMBIGUATION

Word-sense disambiguation (WSD) is the problem of determining in which sense a word, having a number of distinct senses, is used in a given sentence. While the syntactic ambiguity of a given word largely can be resolved in language processing through part-of-speech taggers [Church 1988; Brill 1995; Collins 1996; Ratnaparkhi 1998; Charniak 2000], word-sense disambiguation remains a challenge in the field of statistical natural language processing. In general, word-sense disambiguation includes both *dictionary-based* approaches, in which disambiguation is carried out using information from an explicit lexicon or knowledge base (e.g., matching words in various definitions of the word being disambiguated to the text where this ambiguous word occurs [Klavans et al. 1990]), and *context-based* [Lesk 1986] approaches. In the latter, words are disambiguated by using information gained through training on some corpus or context, rather than an explicit knowledge source. Most approaches, however, have been evaluated only on a small scale of sense disambiguation, limiting the results to theory, and without real applications.

In our study, we have applied context-based disambiguation on a large scale. We have assumed that the content of either the entire abstracts or the local sentences gives consistent cues to the sense of a given abbreviation; therefore, our features include all words in the abstracts or in the local sentences. We have also tested, as an alternative, limiting our features locally to the words of only those sentences that contain the undefined abbreviations. We use as learning features not only single words, but also contiguous word-pairs without punctuation or stop-words. Including word-pairs as a feature has already proven a strong enhancement in text categorization [Wilbur 2000]. We did not apply other features such as word position, morphological information (e.g., capitalization), stop-words, and parts-of-speech, since an earlier study [Hatzivassiloglou et al. 2001] indicated that these add little to disambiguation performance.

Furthermore, we have explored as a feature the medical subject headings (MeSH) terms that were assigned to the abstracts. MeSH terms express a semantic classification of the articles; they are organized hierarchically and multiple terms are typically assigned to each article, and in general, they provide a consistent way to both retrieve information and classify texts [Stapley and Benoit 2000]. The assigned MeSH terms do not often appear in the abstracts, thus they serve as semantic, class-based smoothing factors in our system, as we now discuss.

4. CLASS-BASED SMOOTHING BASED ON MESH TERMS

Smoothing methods are typically used to solve the problem of data sparseness within probabilistic natural-language (such as *n*-gram). Usually, the training set for any statistical learning system is a small subset of the universe; some word sequences or other combinations of features may be missing in the training set. This problem becomes exponentially more acute with an increasing number of variables whose joint distribution is estimated. In fact, for many models, many variable combinations will of necessity not be seen with realistically sized training sets. The missing data leads to “zero probability” in maximum-likelihood estimation. Smoothing techniques are used to adjust the probability estimates of small-count events and to alleviate zero probabilities in the training set. Essentially, they slightly modify the estimated probabilities of seen events (downwards from maximum likelihood) to reallocate the probability mass for unseen events. Smoothing techniques vary from simple *additive* smoothing, in which we pretend that each event occurs once more than it actually does [Church 1988; Lidstone 1992], to *interpolation,* in which probabilities are combined with an estimator that is guaranteed to be nonzero for unseen word-pairs [Jelinek and Mercer 1980], to sophisticated methods that *back-off* the probabilities of individual events to probabilities of larger (nonempty) classes [Pereira et al. 1993; Resnik 1993]. In our study, we have used MeSH terms, which represent semantic categories where the abstract is assigned, hence providing a high-level categorization of the semantic context in which the abbreviation is used. We have compared the performance of our system, both with and without MeSH terms, to measure the effect of this class-based smoothing factor.

To account for unseen full forms, we have applied a *replacement smoothing* method. For each abbreviation, we created a new sense, *XXX*, in order to collectively represent all the unseen senses of undefined abbreviations. *XXX* is a string that is never seen as a valid full form for any abbreviation, and therefore difficult to estimate statistics for the purpose of full-form prediction; and we assigned *XXX* as the full form of this abbreviation. We used six as the minimum frequency threshold for including a full form by itself rather than pooling its occurrences in the *XXX* set. For example, the full form of the abbreviation “ASP” *aggregate shape parameter*, that is, occurred in less than six abstracts, so during training we replaced *aggregate shape parameter* with *XXX* in those abstracts where *aggregate shape parameter* appeared as the full form of "ASP". A parameter of six was chosen as a tradeoff, it is a low enough threshold to enable predictions for many abbreviations, yet sufficiently high to allow reasonably reliable learning.

5. UNSUPERVISED LEARNING THROUGH SUPERVISED METHODS

Typical approaches towards resolving the problem of word-sense disambiguation include both unsupervised (e.g., context-group discrimination [Schuetze 1998]) and supervised (e.g., Bayesian classification [Gale et al. 1992]) methods. Although the latter methods usually outperform unsupervised learning, the production of labeled training data is expensive; this is known as a *knowledge acquisition bottleneck* [Engelson 1996]. Bootstrapping [Hearst 1991] and co-training [Blum and Mitchell 1998] partially address this problem and have been applied successfully in the biomedical domain to enhance the tasks of gene or protein synonym identification [Yu and Agichtein 2003], and named entity recognition [Wellner 2005].

An attractive method was introduced by Yarowsky [1995], namely, to automatically obtain high-quality training data through simple methods. Specifically, the author used a small number of seed-words that were highly correlated with the targeted sense labels. For example, *manufacturing* and *life* are two words that, if present in the context of the ambiguous word *plant*, are strongly correlated with the senses *factory* and *living organism*, respectively. Starting from an unannotated text, he extracted sentences where the seed-words occurred, together with the word being disambiguated, and thusly created a training set automatically, with the sense labels having a high probability of being correct. The training set was then used for supervised learning. [Yarowsky 1995] reported up to a 96% precision rate in disambiguating a number of words having two senses each. [Hatzivassiloglou et al. 2001] applied a similar technique to disambiguate whether a term was a gene, a protein, or RNA. In both systems, the automatically obtained labels can be used for both training and evaluation purposes. In our study, we have applied a similar idea to obtain training and testing sets automatically. However, in contrast with Yarowsky [1995] and Hatzivassiloglou et al. [2001] who applied a handful of seed-words collocating with high precision with a given sense, we automatically identify the senses (i.e., the full forms) of abbreviations with a sophisticated algorithm. Based on whether both the abbreviation and the full form appear in the same abstracts, we automatically create a training set and testing set, respectively.

6. ASSUMPTIONS OF OUR APPROACH

To obtain the dictionary of paired abbreviations and full forms, we assumed that the complete set of MEDLINE abstracts was a relatively exhaustive resource of paired abbreviations and full forms in the biomedical domain, that the authors defined the abbreviations (i.e., full forms of abbreviations appear in the abstracts) when the abbreviations were newly introduced in the literature, and that the literature was redundant (i.e., the full forms appear in multiple abstracts). To disambiguate abbreviations, we assumed that the full forms of an abbreviation represented its senses. We assumed that MeSH term assignments were consistent. We also assumed *one sense per collocation* (i.e., nearby words provide strong and consistent clues to the full form of an abbreviation) and *one sense per discourse* (i.e., the full form of an abbreviation does not change within the abstract). The concepts of one sense per collocation and one sense per discourse were introduced by Church and Gale [1991] and Yarowsky [1995].

We have empirically evaluated the one sense per discourse hypothesis in our study. We extracted abbreviation-full form pairs from eleven million MEDLINE records (titles and abstract bodies taken together; years 1966–2001). We identified abbreviations associated with two or more different full forms within the same MEDLINE record. For this part of our study, we assumed that the sense of an abbreviation does not change until a new sense is defined. We found 33,038 MEDLINE records that defined abbreviations two or more times. Since MEDLINE records use an XML format to separate the title from the abstract body, we found that these records all defined abbreviations differently between their titles and abstract bodies. For example, Turteltaub and Dingley [1998] defined the abbreviation AMS as *accelerated mass spectrometry* in the title and *accelerator mass spectrometry* in the abstract body. We applied the methods we will describe in Section 7, Step 3, and found that the full forms associated with the same abbreviations in the same abstracts are synonyms. We therefore concluded that our data largely follows the one sense per discourse principle. The rest of the assumptions were empirically tested and the results are reported next.

7. DATA COLLECTION, TRAINING, AND TESTING SETS

We have evaluated our disambiguation methods with MEDLINE abstracts. MEDLINE is a publicly accessible database[1] maintained by the National Library of Medicine (NLM) that contains bibliographic citations and author abstracts from more than 4,800 biomedical journals published in the United States, as well as 70 other countries. The database currently holds over 15 million citations dating back to 1966.[2] We focused on MEDLINE because it is a freely-distributed large corpus.

To train our system and evaluate our methods, we randomly selected 60 biomedical abbreviations (denoted as $\mathrm{S_{abb}}$) from the Unified Medical Language System (UMLS) SPECIALIST lexicon, which consists of over 10,000 common biomedical abbreviation-full form pairs [McCray 1998]. For each abbreviation, we extracted from the complete collection of MEDLINE abstracts any that contained it (in either defined or undefined form). The total number of MEDLINE abstracts from 1966–2001 that contain any of the 60 abbreviations is 152,768. This collection (denoted A_T) was used for both training and testing. The number of abstracts in A_T that have assigned MeSH terms is 147,866 (some recent abstracts were in the process of having them assigned). We obtained

[1] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi, accessed on Dec. 1, 2001.

[2] Note that the MEDLINE records we used for this study consist of 12 million citations from 1966–2001.

training and testing sets from A_T through the following four steps: preprocessing, extraction of abbreviations-full form pairs, recognition of synonymous full forms, and extraction of training and testing sets on the basis of whether the full forms appear in the abstracts.

Step 1. *Preprocessing.* We preprocessed A_T by first converting the format of the abstracts from XML to regular text files, lowering their cases and removing hyphens: A hyphen was removed when it had been used to break-up a line; it was replaced with a space when used to connect multiple words. We then parsed the abstracts into sentences based on a set of handcrafted rules. The most challenging task for splitting a sentence in the biomedical domain is that of correctly identifing abbreviations, which are frequently used. We have summarized a set of heuristic rules for identifying a given abbreviation and have incorporated them to parse abstracts into sentences. Specifically, we consider a word so as an abbreviation if it matches to a list of common biomedical abbreviations, or the word is a one-letter word (a–z, but not "I"), or the word has no vowels and no digits, or the word contains a period and has no digits. We have evaluated the sentence splitter with a total of 70 MEDLINE abstracts that were manually parsed into sentences and we found a performance of 92.2% for precision and 94.5% for recall. The performance is a little better than that of MXTerminator [Ratnaparkhi 1998], which achieved 91.9% precision with 93.4% recall.

Step 2. *Extraction of Paired Abbreviations and Full Forms.* We applied AbbRE to the preprocessed set of MEDLINE abstracts A_T to obtain a dictionary of abbreviation-full form pairs. We chose AbbRE over other systems for mapping defined abbreviations to full forms because AbbRE was developed and evaluated specifically in the biomedical domain with good precision. AbbRE's output on A_T is a dictionary of paired abbreviations and full forms of the abstracts. From the output, we only took those abbreviations belonging to the set of 60 that we studied.

Step 3. *Full Form Normalization.* Term variations are abundant in the biomedical domain. To be precise, one meaning of an abbreviation can frequently come with different full forms, each incorporating variations that either derive from the same root, or have a number of compound forms or diverse additions to the basic words that are being abbreviated. It is important to normalize these full forms that represent the same sense, otherwise, these full forms would represent "distinguished senses" that only introduce noise into training and testing. We carry out three processing steps to normalize full-form variations:

(1) In the first step, Porter's stemming algorithm [Porter 1980] is applied to all the full forms that are found for a single abbreviation. Full forms stemming from the same string are identified as synonyms.

    Example. For the abbreviation "scid," the forms
    - —severe combined immunodeficiencies;
    - —severe combined immunodeficiency;
    - —severe combined immunodeficient;
    - —severely combined immunodeficient; and
    - —severe combine immunodeficient

all stem under Porter's [1980] algorithm to the single form "sever combin immunodefici," which allows us to conclude that they are variations. If two full forms do not produce the same stemmed form, we then apply two tests to measure their similarity: a high degree of similarity indicates that they represent the same full form.

(2) The first test is based on the Trigram matching algorithm [Wilbur and Kim 2001]. We apply this in its cosine form with IDF global and log(TF+1) local weights. Here, *TF* is the count of the trigram that is being weighted within a phrase. This algorithm, when applied to two phrases, produces a score in the range 0–1 as a rating of their similarity. We found that when the stemmed versions of two phrases represent synonyms and both phrases are significantly longer than the abbreviation, they generally achieve a Trigram similarity score of 0.5 or greater. However, if one of the phrases is short, the threshold may be set higher. This has led us to an empirical formula for calculating the threshold below which we do not consider the phrases to be candidates for synonymy. If *len* is the length of the shorter of two phrases and len_{abr} is the length of the abbreviation, then the score must be at or above a threshold of $0.5 + 0.5\exp(-len - 1 + 2len_{\mathrm{abr}})$.

Example. Again, for the abbreviation "scid," the strings "b icr scid scid" and "scid fc" achieve a trigram similarity score of 0.651908. This relatively high score is a known problem for a cosine similarity calculation applied to short documents (in our case, short strings). It requires a correction [Singhal et al. 1996], and our approach is to use the aforementioned formula to produce a threshold that is higher for short strings. In this case, the threshold is 1.0 and excludes the pair of strings from further consideration for synonym status.

(3) While the processing under step (2) eliminates the bulk of phrase pairs that are not synonyms, we still find it necessary to apply one additional test in order to further eliminate nonsynonymous phrase pairs. Now, we look for a consistent assignment of abbreviation letters to the two phrases being considered. If there is a consistent assignment, the two phrases are accepted as a valid synonym pair, otherwise they are rejected.

Let "abr" denote an abbreviation of length len_{abr} and $phr1$ denote a corresponding full form of length $len1$. Then, a model for $phr1$ is an integer array $mod1$, of length len_{abr}, taking values between 0 and $len1$ and satisfying the following:

$$\text{For each } i \text{ and } j, 0 \le i < j < len_{abr} \text{ we require } \mathrm{mod}\,1[i] < \mathrm{mod}1[j]$$
$$\text{and for any } i,\ 0 \le i < len_{abr} \text{ we require } abr[i] = phr1[\mathrm{mod}1[i]].$$

Thus, a model is just an identification of the letters of the abbreviation with the same letters in the phrase, in the same order. A model is one way to construct the abbreviation. Of course, not all models are of equal quality or have the same likelihood of explaning the abbreviation. This leads us to score a given model as to its quality. Each letter assignment of the model is scored. The score for *mod*1[i] is 1 if *mod*1[i] is 0 or if *phr*1[*mod*1[i]-1] is other than an alphabetic character (usually a space, but possibly a dash, number, etc.). Otherwise, the score is 0. The score for the model is the sum of scores for each of the characters it maps. We refer to a model having the highest possible score as *optimal*. Any phrase that possesses a model will have one or more optimal models.

As an example, consider the abbreviation “acd” and its stemmed full form “acid citrat and dextro.” A model will consist of an assignment of the letters of the abbreviation to letters in the full form that occur in the same order. Thus, “$\underline{a}$cid $\underline{c}$itrat an$\underline{d}$ dextro” would denote a model with a score of 2, as does “$\underline{a}\underline{c}$id citrat and $\underline{d}$extro.” Neither of these models is optimal because there is a model “$\underline{a}$cid $\underline{c}$itrat and $\underline{d}$extro,” which achieves a score of 3 and is in fact optimal. It is thus readily apparent that the optimal model represents the origin of the abbreviation. Hence, we see the importance of optimal models.

Now suppose *mod*1 is an optimal model for *phr*1 and *mod*2 is the same for *phr*2, both based on the abbreviation “abr”. In order to consider these two models to be consistent, we require that certain conditions be satisfied for each letter of the abbreviation. For any i, $0 \leq i < len_{abr}$, let $e1[i]$ denote the number of characters from $phr1[mod1[i]]$ to the next nonalphabetic character, to the end of $phr1$, or to $phr1[mod1[i{+}1]]$ if it has been defined, (whichever is the smallest). In like manner, define $e2[i]$ for $phr2[mod2[i]]$. If $0 \leq i < len_{abr}$, then we know that $phr1[mod1[i]]$ and $phr2[mod2[i]]$ are the same. We require that the agreement

$$phr1[\mathrm{mod}1[i] + j] = phr2[\mathrm{mod}2[i] + j]$$

continues for $0 < j \leq \min(e1[i], e2[i])$. Additionally, we require that

$$\min(e1[i], e2[i]) \leq 2 \text{ implies} \max(e1[i], e2[i]) \leq 2$$

for any i, $0 \leq i < len_{abr}$.
This states that if the shorter of two segments to be matched is not more than two characters, then the longest segment cannot be more than two characters either. This particular condition is imposed because very short segments tend to be highly ambiguous and should only be allowed to identify with other highly ambiguous segments. If these conditions are satisfied, then we say that *mod*1 and *mod*2 are consistent. If two phrases have consistent optimal models, then they pass the consistency test for acceptance as synonyms. Conversely, if either fails to have a model or they simply fail to have a pair of consistent optimal models, they fail the consistency test.

Example. Again consider the abbreviation “acd” and two stemmed full forms “acid citrat dextro formula a” and “acid sodium citrat dextro.” These latter have optimal models “$\underline{a}$cid $\underline{c}$itrat $\underline{d}$extro formula a” and “$\underline{a}$cid sodium $\underline{c}$itrat $\underline{d}$extro,” and it thus becomes readily apparent that when corresponding marked letters are compared, the parts of words following these marked letters are the same and the models are therefore consistent. The full forms are hence considered synonyms.

Step 4: Extraction of Training and Testing Sets Based on Whether the Abbreviations are Defined in the Abstracts. For each abbreviation, we selected training and testing abstracts from a total of 152,768 abstracts (A_T). The training abstracts each contain at least one abbreviation in association with a defining full form. Testing abstracts include abbreviations only. The average number of training and testing abstracts for any abbreviation (with their standard variations) are 781 ± 807 and 1,456 ± 1,831, respectively.

8. MACHINE-LEARNING SYSTEMS

We have applied two supervised machine-learning methods, namely, naïve Bayes and support vector machines (SVMs); both are widely used for text categorization problems. The naïve Bayesian algorithm assumes that attributes are distributed independently within the classes to be learned. Each attribute (in this case, a term) can thus be weighted separately, based on its distribution in the training set. We score documents in the test set by summing the weights of the terms they contain. For details, the reader may consult Langley et al. [1992], Langley and Sage [1994], and Wilbur [2000]. We have found that this learner's performance is improved if we remove from the scoring any terms with weights of less than 1.0 and we refer to this as feature selection (weight >1.0). Such feature selection (weight >1.0) is used implicitly in this study for all applications of the naïve Bayes algorithm.

Support vector machines (SVMs) learn the linear hyperplane that separates a set of positive examples from a set of negative with maximum margin (the margin is defined by the distance of the hyperplane to the nearest positive and negative examples). Support vector machines are important because they have improved upon the performance of naïve Bayes in a number of learning environments [Wilbur 2000]. We use Platt's [1999] sequential minimal optimization method of training support vector machines. We have followed the author in taking the error tolerance *eps* to be 0.01. When the algorithm comes within *eps* of satisfying the Karush-Kuhn-Tucker conditions, it halts with the resultant solution. There is a second parameter C, which is an upper bound on the individual Lagrange multipliers in the problem. Its interpretation is a penalty factor that puts sanctions on the objective function we seek to maximize. For each data point that violates the margin, the penalty is C times the distance of the violation. Values of 1.0, 0.5, 0.1, and 0.05 for C have been tested [Wilbur 2000] and all gave close to the same result, with 0.1 and 0.05 as the best and essentially equivalent. The value of 0.01 gave a decreased performance by 0.5%. In the work reported here, we used a C value of 0.1. In order to obtain efficiency in SVM training, we pruned the set of features by using a chi-square criterion. Each feature is assigned chi-square values coming from the contingency table that relates relevant to nonrelevant and feature-present to feature-absent sets. We retained features with chi-square values greater than 3.84. The chi-square criterion has been found useful for feature selection [Yang and Pedersen 1997] and we have chosen a value of 3.84 to provide features with a correlation to the training data that is significant at the 0.05 level [Wilbur 2000].

9. LEARNING FEATURES

The learning features included single words, contiguous word-pairs without either punctuation or stop-words [Wilbur 2000], and the MeSH terms assigned to the abstracts. Note that we removed stop-words and performed no stemming.

We did the learning study once with the full forms present in the text. We subsequently repeated the study with the full forms removed from the training documents, but retained this information for evaluation purposes. In both studies, we compared the performance of the following five feature sets:

(1) The MeSH terms that were assigned to the abstracts in addition to the words and word-pairs that were present in the abstracts;
(2) both words and word-pairs that were present in the abstracts;
(3) the MeSH terms that were assigned to the abstracts, as well as the words and word-pairs in sentences containing the abbreviations;
(4) words and word-pairs in sentences in which the abbreviations appear; and
(5) the MeSH terms assigned to the abstracts.

10. CROSS-VALIDATION AND PREDICTION

To describe our means of cross-validation and prediction, we use the abbreviation "abl" as an example. Let $A_T^+(abl)$ represent all those abstracts from A_T that have the abbreviation "abl" and one of its full forms, and likewise let $A_T^-(abl)$ represent all the abstracts that have "abl" but not one of its full forms. There are several full forms of "abl", such as *agaricus bisporus lectin* and, *abetalipoproteinemia*. Let us suppose there are M full forms for "abl" and denote them by the list $\{ff_i\}_{i=1}^M$. Because naïve Bayesian and support vector machine-learning methods require a reasonable number of examples on which to train, we require a full form to occur at least six times in association with its abbreviation in order to include it on the list $\{ff_i\}_{i=1}^M$. If the frequency of a full form is less than six, we put it into the category *XXX*. We treat *XXX* as a miscellaneous category and also include it on the list. The number of abstracts belonging to the category *XXX* must also be at least six. If this condition cannot be satisfied, the abbreviation is to be excluded from consideration.

Given a full form ff_k, the set $A_T^+(abl)$ divides naturally into the two sets, $A_T^+(abl, ff_k)$ and $A_T^+(abl, \sim\ ff_k)$, that, respectively, do and do not contain the full form ff_k. Training involves learning to distinguish $A_T^+(abl, ff_k)$ from $A_T^+(abl, \sim ff_k)$. All of our training involves cross-validation. We generally perform n-fold cross-validation for some small n. This involves randomly dividing each set $A_T^+(abl, ff_k)$ and $A_T^+(abl, \sim ff_k)$ into n disjoint, and as nearly equal pieces as possible. We use an n of 10 where possible. When there are insufficient training examples (i.e., $6 \leq n < 10$), we used n. The attempt at equal division here is to prevent the pathological situation when some training set has no good or no bad abstracts and one cannot learn. This is potentially a problem when there are few good abstracts. Then, in n cross-validation rounds we learn the difference between good (associated with the full form) and bad (not associated with the full form) from the training set and then apply this learning to produce scores for the abstracts in the test set or sets. This is done for each of the full forms.

Now, our objective is to assign each abstract to some full form. Because there are many full forms and we cannot directly compare the scores produced based on the learning for different full forms, we must convert the scores to probability predictions that a given full form is correct for a particular document. Such probabilities can be directly compared across multiple predictions. This leads us to approach the problem of prediction in two stages.

Stage (1). We perform tenfold cross-validation on $A_T^+(abl, ff_k)$ and $A_T^+(abl, \sim ff_k)$. All the scores on the test data from all ten rounds are combined and the pool adjacent violator (PAV) algorithm [Ayer et al. 1954; Hardle 1991] is applied to learn the relationship between a given score and the probability of having ff_k as the correct full form. For a description of the PAV algorithm, see the Appendix. In some cases, the number of documents having ff_k as their full form are too few to allow for tenfold cross-validation. In these cases we still perform at least a fivefold cross-validation to learn the relationship between score and probability. This is guaranteed by the requirement that at least six documents have ff_k as their full form.

Stage (2). We perform tenfold cross-validation on $A_T^+(abl, ff_k)$ and $A_T^+(abl, \sim ff_k)$. In each of the ten rounds, the learning on the training data is used in two ways:

First, the test documents in both $A_T^+(abl, ff_k)$ and $A_T^+(abl, \sim ff_k)$ are scored and the scores are converted to probabilities for whether ff_k is the correct full form. Second, all the documents in $A_T^-(abl)$ are treated in the same manner.

The conversion from scores to probabilities is done using the results of Stage (1). When the tenfold cross-validation is completed each member of $A_T^+(abl)$ will have received a probability predicting its likelihood of having ff_k as its correct full form exactly once. Similarly, a document in $A_T^-(abl)$ will have received a probability predicting its likelihood of having ff_k as its correct full form ten times. These ten probabilities are averaged to yield a single result.

The preceding procedure we have described in two stages applies to a single full form for the abbreviation "abl". This same plan is followed for each of the M full forms. The end result is a probability vector predicting the likelihood that an abstract belongs to the full form ff_k for each of the M full forms. This applies to all abstracts in $A_T^+(abl)$ and $A_T^-(abl)$. Finally, we assign as our prediction to each abstract the full form that has the highest probability. The correctness of the predictions on $A_T^+(abl)$ is assessed by a comparison with the known full forms, while correctness of predictions on $A_T^-(abl)$ is assessed by comparison with the human judgments as described in the next section.

11. EVALUATION METHODS

Since we introduced *XXX* to represent unseen full forms, we performed the evaluations through *coverage* (i.e., the percentage of all predictions that predict non-*XXX*) and *precision* (i.e., the number of correctly predicted full forms divided by the total number of predicted full forms). Coverage values indicate the capability of our system for predicting non-*XXX* full forms. We introduced three precision measures: *Prec_1, Prec_2*, and *Prec_3*. P*rec_1* is the number of correct non-*XXX* predictions divided by the total number of non-*XXX* predictions. *Prec_2* is the total number of correct *XXX* predictions divided by the total number of *XXX* predictions. *Prec_3* is the total number of correct predictions divided by the total number of predictions. We also calculated *macro-precision* measures (i.e., the average precision across different abbreviations).

12. GOLD STANDARD

We selected a total of 300 MEDLINE abstracts that incorporate the 60 abbreviations we would use for evaluating our automatic biomedical disambiguation. We divided the 60 abbreviations ($\mathrm{S}_{\mathrm{abb}}$)into two subsets (denoted as $S1$ and $S2$), consisting of 10 and 50 abbreviations, respectively. We randomly selected 15 abstracts from the test set ($T2$) for each abbreviation in $S1$, and 3 from T_2 for each in S_2. The total number of evaluated abstracts was 300 ($10 * 15 + 50 * 3$). Our approach to selecting abstracts attempted to strike a balance between the number of abstracts for each abbreviation and the total number of evaluated abbreviations, while at the same time minimizing the total number of abstracts that require expert evaluation.

To generate a gold standard, we first selected 100 abstracts ($10 * 5 + 50 * 1$) that balanced both the number of abstracts and abbreviations. Three authors of the present article, namely, Dr. Kim, Dr. Wilbur, and Dr. Yu annotated them independently to identify the full forms of the abbreviations in these abstracts, and did so by any means. For example, they could utilize outside resources such as full-text articles, the Worldwide Web, textbooks dictionaries, and consults with other biomedical experts.

The three researchers then calculated the overall agreements amongst themselves. We independently identified five abbreviations out of the total of 100 abstracts that are not true abbreviations, but represent common English words (e.g., "apt" in the MEDLINE abstract pmid = 2061735), and therefore excluded these five abbreviations. Our agreement calculations were subsequently based on the remaining 95 abstracts.

There were four instances in which the annotators selected "unknown" to represent abbreviations of unidentifiable full forms. We had to consider the term "unknown" as a full form and "unknown" did not match any known full form. The results of the pairwise agreements are listed in Table I. We found that the three authors agreed on the full forms for 61 (or 64.2%) of the evaluated abbreviation occurrences, and for 83 (or 87.4%) of abbreviation occurrences, at least two evaluators were in agreement. The results suggest that although the number of our pairwise agreements was not high, the disagreements tended to correlate with certain abstracts that may have represented the "harder" cases of mapping abbreviations to full forms.

The three authors then openly discussed the cases of dissagreement and found that a consensus was easily reached when one of the annotators could supply the "evidence" for the correct full form. These forms of evidence were frequently other related abstracts that incorporated the predicted full forms.

The remaining 200 abstracts were annotated by Dr. Kim and Dr. Wilbur. Excluding abstracts where the abbreviations were common English words,

Table I. Pairwise Overall Agreements on Identifying Full Forms

	Dr. Kim	Dr. Wilbur	Dr. Yu
Dr. Kim	–	76.8%	68.4%
Dr. Wilbur	–	–	70.5%
Dr. Yu	–	–	–

Agreements were among three annotators: 1 Dr. Kim, Dr. Wilbur, and Dr. Yu. The full forms were for specified abbreviations in 100 MEDLINE abstracts.

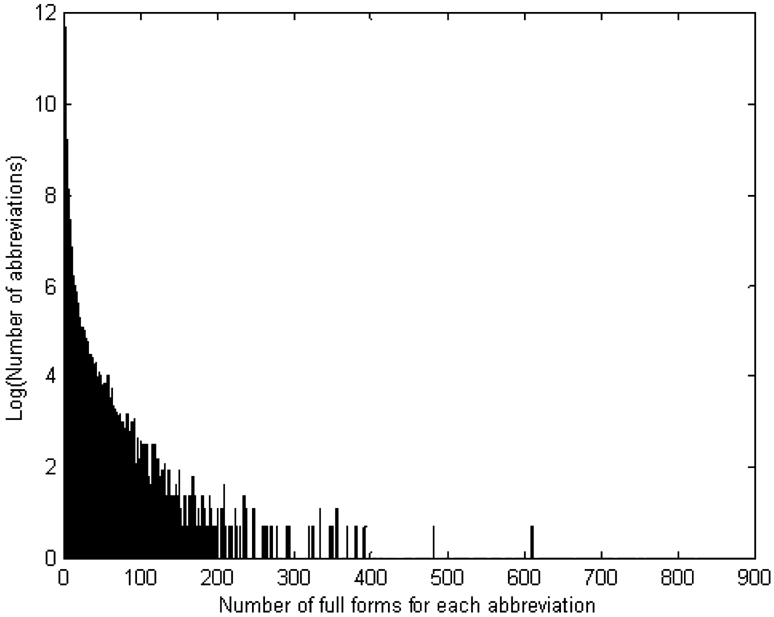

Fig. 1. Distribution (from 11 million MEDLINE records) of the number of abbreviations paired with different full forms.

misspelled (e.g., "emb" should have been "emg" in abstract pmid = 226706), or abbreviations for which none of the annotators could identify full forms (e.g., "rvh" in the abstract pmid = 1354400), we calculated a total of 281 abstracts with the abbreviations that had been annotated to have associated full forms.

13. ABBREVIATION-FULL FORM DICTIONARY

Our AbbRE has extracted a total of 3,287,178 unique abbreviation-full form combinations in 11 million MEDLINE records. Figure 1 plots the distribution of the number of abstracts and abbreviations in the 11 million MEDLINE records. The figure shows that from 11 million MEDLINE records, there were a few abbreviations which had a large number of full forms (>100), but most a modest number. We found an average of 9.6 with a standard deviation of 3.7. Highly ambiguous abbreviations (i.e., number of full forms exceeding 600) were: *ap, ca, cd, cp, cr, cs, ct, pa, pc, pi, ps, sc,* and *sp*.

```
1377599  0.0    c termin peptid
1377599  0.0    citrat transport protein
1377599  0.0    beta carboxi termin peptid
                carboxi termin peptid
1377599  0.0    beta carboxytermin peptid
                carboxyl termin peptid
                carboxytermin peptid
1377599  0.17   xxx
1377599  0.91   cytidin guanosin and uridin triphosph
                cytidin triphosph
                cytidinetriphosph
                cytidine triphosphate
```

Fig. 2. A naïve Bayes sample prediction. The abstract pmid is 1377599; 0 – 1 represents predicted probabilities of full forms (e.g., *c termin peptid* and *citrat transport protein*). Note that the full forms are stemmed. Synonymous full forms are listed as one cluster, to which the probability is assigned. The full form(s) with the highest probability is selected as the predicted full form(s) for an abbreviation.

We extracted a total of 80,998 unique abbreviation-full form pairs from the 152,768 (A_T) MEDLINE records in our study; when we limited our pairs only to the 60 abbreviations in our study, we obtained a total of 567 unique abbreviation-full form pairs. The average number of full forms for a given abbreviation within our training set was 9.44 with a standard deviation of 14.45. The baseline precision obtained by assigning all abbreviations to their most frequent full form within our training set (*Prec_1*) was 0.31 with a standard deviation of 0.21.

14. A PREDICTION EXAMPLE, CROSS-VALIDATION, AND PREDICTION RESULTS

Figure 2 shows a prediction sample. The results of cross-validation and prediction are listed in Table II. We plotted the distributions of the number of abstracts and full forms (i.e., the ambiguity) (depicted in Figure 3); the number of abstracts and *Prec_1* (shown in Figure 4); and the number of full forms and *Prec_1* (in Figure 5) for every abbreviation in the training set.

We applied bootstrapping analyses and the results (at the alpha = 0.05 level) show that naïve Bayes and SVMs performed equally well in all cases of cross-validation (*Prec_1*) regardless of whether the full forms were present, though the precision of SVMs was generallly higher than that of naïve Bayes. Naïve Bayes performed differently than SVMs in predicting *XXX* (*Prec_2*). When full forms were present, in most cases (except when MeSH terms were used as the only feature) naïve Bayes outperformed SVMs. On the other hand, when the full forms were absent, naïve Bayes and SVMs were statistically indistinguishable from one another in predicting *XXX*. The coverage ranged from 0.53 to 0.75.

15. DISCUSSION

We have developed and evaluated a context-based unsupervised machine-learning approach for disambiguating biomedical abbreviations. We first automatically obtained for a training set those abstracts that incorporate

Table II. Results of Cross-Validation and Prediction of Full Forms by Naïve Bayes and SVM

				Results			
Learning Algorithm	Learning Features			Coverage	Prec_1	Prec_2	Prec_3
NB	Abstract+MeSH	+	C1	0.80	0.87	0.81	0.87
			C2	0.84	0.93	0.84	0.93
			P	0.65	0.79	0.83	0.80
		−	C1	0.80	0.86	0.79	0.86
			C2	0.84	0.92	0.83	0.92
			P	0.65	0.79	0.83	0.80
	Abstract	+	C1	0.80	0.88	0.83	0.88
			C2	0.84	0.93	0.85	0.93
			P	0.63	0.78	0.80	0.79
		−	C1	0.79	0.87	0.80	0.86
			C2	0.83	0.93	0.82	0.92
			P	0.64	0.77	0.79	0.78
	Sentence+MeSH	+	C1	0.79	0.89	0.82	0.88
			C2	0.84	0.94	0.84	0.93
			P	0.62	0.80	0.82	0.80
		−	C1	0.78	0.88	0.78	0.87
			C2	0.83	0.93	0.81	0.92
			P	0.63	0.79	0.84	0.81
	Sentence	+	C1	0.78	0.90	0.82	0.89
			C2	0.82	0.94	0.82	0.93
			P	0.56	0.77	0.72	0.75
		−	C1	0.78	0.89	0.78	0.87
			C2	0.83	0.93	0.76	0.91
			P	0.58	0.77	0.74	0.75
	MeSH	+	C1	0.79	0.81	0.66	0.80
			C2	0.83	0.77	0.69	0.77
			P	0.74	0.67	0.93	0.74
SVMs	Abstract+MeSH	+	C1	0.79	0.93	0.88	0.93
			C2	0.83	0.96	0.88	0.95
			P	0.59	0.71	0.69	0.70
		−	C1	0.79	0.92	0.85	0.91
			C2	0.83	0.95	0.87	0.95
			P	0.57	0.74	0.77	0.76
	Abstract	+	C1	0.79	0.94	0.86	0.93
			C2	0.79	0.94	0.88	0.93
			P	0.53	0.78	0.72	0.75
		−	C1	0.78	0.92	0.83	0.91
			C2	0.83	0.95	0.84	0.94
			P	0.56	0.72	0.76	0.74
	Sentence+MeSH	+	C1	0.78	0.95	0.85	0.93
			C2	0.83	0.97	0.86	0.95
			P	0.54	0.84	0.76	0.80
		−	C1	0.77	0.93	0.83	0.91
			C2	0.83	0.96	0.83	0.94
			P	0.55	0.82	0.77	0.80
	Sentence	+	C1	0.77	0.95	0.83	0.93
			C2	0.82	0.96	0.81	0.95
			P	0.50	0.78	0.69	0.73
		−	C1	0.77	0.92	0.79	0.90
			C2	0.82	0.95	0.78	0.93
			P	0.54	0.76	0.72	0.74
	MeSH	+	C1	0.78	0.81	0.67	0.79
			C2	0.83	0.89	0.65	0.86
			P	0.73	0.74	0.93	0.79

Abstract+MeSH: Words in abstracts and the MeSH terms assigned to abstracts are learning features.
+ and − : the presence and absence, respectively, of full forms in the training documents.
*C*1: cross-validation results without full-form normalization.
*C*2: cross-validation results with full-form normalization.
P: prediction results with full-form normalization.
Cover: percentage of full forms predicted non-*XXX*.
Prec_1: percentage of correct non-*XXX* predictions.
Prec_2: percentage of correct *XXX* predictions.
Prec_3: percentage of all correct predictions.

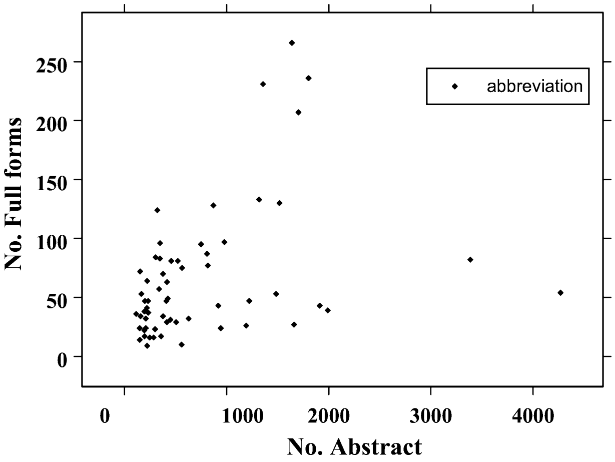

Fig. 3. A plot of the number of abstracts versus the number of full forms for every abbreviation in the training set.

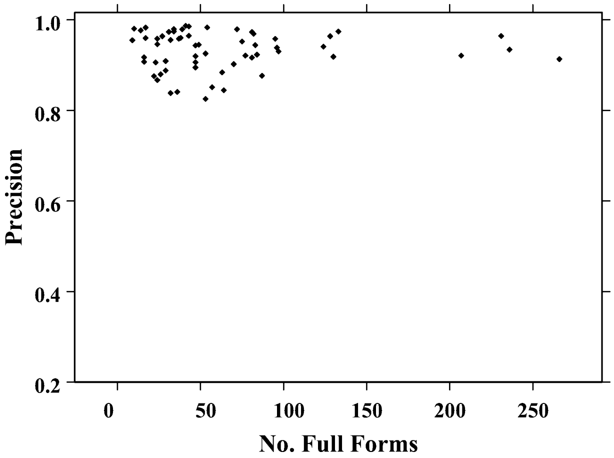

Fig. 4. Relation between naïve Bayes cross-validation precision (Prec_1, learning features: sentence+MeSH) and the number of full forms for every abbreviation in the training set.

biomedical abbreviations and full forms. We applied supervised machine-learning algorithms (naïve Bayes and SVMs) to learn from the training sets so that we could automatically predict the full forms of undefined biomedical abbreviations in MEDLINE abstracts. Our model applies unsupervised approaches through supervised learning algorithms.

The core difference between our model and previous ones [Yarowsky 1995; Hatzivassiloglou et al. 2001] lies in the fact that we did not depend on expert knowledge or an existing dictionary for listing senses; instead, we obtained senses automatically from MEDLINE. We have also developed advanced approaches to normalize full-form variations. In addition, we disambiguated an average of ten "senses" for each abbreviation. Since we automatically assigned

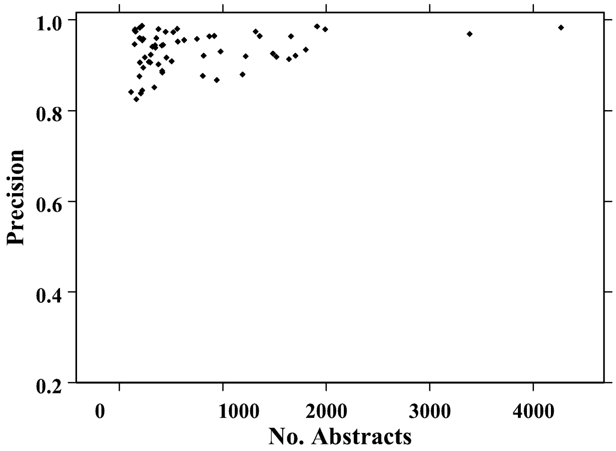

Fig. 5. Relation between naïve Bayes cross-validation precision (Prec_1, learning features: sentence+MeSH) and the number of abstracts for every abbreviation in the training set.

full forms as "senses," we sometimes faced the problem of differentiating terms with related meanings. For example, we disambiguated "accelerated cell death" from "apoptotic cell death," two processes that are closely related, though not identical. Generally, other disambiguation studies have dealt with fewer meanings and meanings which are also quite distinct.

We measured the performances with three precision values namely, *Prec_1*, *Prec_2*, and *Prec_3*. Recall that *Prec_1* is the number of correct non-*XXX* predictions divided by the number total of non-*XXX* predictions; *Prec_2* is the number of correct *XXX* predictions divided by the total number of *XXX* predictions; and *Prec_3* is the number of correct predictions divided by the total number of predictions. Our results show that *Prec_1*, in general, is higher than *Prec_2*. This is not surprising because *XXX* is the full form we created to overcome the problem of data sparseness.

Our results generally show a significant performance improvement (3∼5% in coverage and 1∼6% in precisions) after the process of full-form normalization. The cross-validation results after full-form normalization show 82∼84% in coverage and 72∼96% in precision.

Our results also show 50∼73% coverage, and 67∼93% precision in prediction. Since our baseline was 31% accuracy, the results show a significant improvement in identifying the full forms of abbreviations. Note that both the coverage and precision were higher in cross-validation than in the actual prediction. This drop in performance is most likely due to the fact that the training and testing sets had different distributional characteristics. Distributional differences may be caused by several factors, including the possibility of missing some senses in a particular corpus, but are more likely due to an inherent characteristic of our approach: There are likely additional factors that prompt an author to define an abbreviation in certain contexts and leave it undefined in others, so the distributional characteristics within the context of defined (training set) and undefined (test set) abbreviations are somewhat different.

The outcome of our research shows little performance differences when different machine-learning algorithms (i.e., naïve Bayes or SVMs) are applied at cross-validation. However, the prediction results of naïve Bayes generally outperform those of SVMs, with differences of 1∼10% in coverage, and 1∼10% in precision. We speculate that performance differences are also due to the dissimilarities of distributional characteristics between training and testing corpora. Naïve Bayes has been shown in our study to be more robust than SVMs. In naïve Bayes, each feature makes its contribution independently, while in an SVM, prominent features in the training data may have the effect of hiding others, which subsequently do not appear. Then, in the testing data, if the most prominent features in training are missing, less prominent ones may not be able to fill in for them because the latter received very low weights.

Our results do not show significant performance differences when the full forms are removed as the learning features. The results indicate that other words or MeSH terms are sufficient for the purpose of abbreviation disambiguation. On the other hand, we have found that different features play various roles for disambiguating abbreviations.

Specifically, our research shows that MeSH terms are important features for predicting the full forms of abbreviations. MeSH terms alone have the highest prediction value in both coverage and precision (71% and 80%, respectively, with naïve Bayes). These results support our hypothesis that MeSH terms serve as semantic smoothing factors and therefore have a positive effect on performance.

In Figure 2, the sum of the predicted full-form probabilities for the abbreviation "*ctp*" is greater than one. Because our data is not perfectly conditionally independent, the implication is that predictions will not be perfect. The fact that we separately calculated each abbreviation suggests that the prediction probabilities will not likely sum up to one. Theoretically, we could calculate posterior probabilities for all the full forms simultaneously and then the predictions would sum up to one, but this would impose significant computational overhead with little chance of gain for our purposes (selecting the most likely full form).

Figure 3 indicates that our data is not homogeneous: The number of abstracts as well as the number of full forms varies among different abbreviations. The plot shows the trend that as the number of abstracts for an abbreviation increases, the ambiguity level also increases, although a few abbreviations that appear very frequently (in >3,000 abstracts) show decreased ambiguity levels. This suggests that common abbreviations are so familiar that people avoid using them in new settings.

Although our evaluation sets consisted of 60 abbreviations with different ambiguity levels and sizes of training sets, Figures 4 and 5 show that the precision for predictions is similar among the various abbreviations. In general, when as the training size increases, the ambiguity level usually increases as well, and therefore, the performance gain with increased training size will tradeoff with the increased ambiguity. Therefore, precision is a complex value that does not simply increase or decrease with training size or ambiguity level.

16. CONCLUSION AND FUTURE WORK

We have developed context-based machine-learning word-sense disambiguation approaches to resolve ambiguous biomedical abbreviations. We generated training sets automatically, based on the explicit link between an abbreviation and its full forms in some documents. We experimented with word features as well as medical subject headings (MeSH) as an external knowledge source. We found that including MeSH terms as learning features enhanced disambiguation. We also found that full-form normalization significantly enhances performance. By combining different learning features and methods, we obtained an overall precision of 74–81% with 53–79% coverage in an independent test set. The best disambiguation results are 79% in coverage, and 80% in precision, obtained by using naïve Bayes trained on MeSH terms with full-form normalization.

In future work we will implement our disambiguation in real systems, such as information extraction and retrieval, and question answering. For example, to answer the question "What is *ctp*?" our disambiguation system may let the user narrow down the range of possible answers to a particular full form and then retrieve the documents for this sense. In addition, the most likely full forms for an abbreviation supplied by the user can automatically be proposed by our system, aiding in query refinement and increasing the quality of the retrieval process. The methods described in this article can also be used to guide retrieval when the user supplies a full term rather than an abbreviation; the documents in which a particular abbreviation appears can be indexed under both the abbreviation and the (automatically determined) appropriate full form, further aiding retrieval.

APPENDIX: The Pool Adjacent Violators Algorithm

Consider the set of data points $\{(sco_i, n_i)\}_{i=1}^{N}$ produced by a machine-learning method, as described in Section 7.4. Here, it is understood that for each i, sco_i is the score produced by the machine-learning method applied to the i-th abstract and $n_i = 0$ or 1, depending on whether the i-th abstract belongs to $A_T^+(abl, \sim ff_k)$ or to $A_T^+(abl, ff_k)$, respectively, for the full form ff_k of interest. Without loss of generality, we may assume these points are arranged in order by the first coordinates, so that sco_i is a nondecreasing function of the index i. In our case, it is appropriate to assign to all points an initial weight of 1.0. Thus we begin by setting

$$p_i = n_i, w_i = 1, i = 1, 2, \ldots, N. \tag{1}$$

Then, the maximum likelihood estimates are $pr[sco_1], pr[sco_2], \ldots, pr[sco_N]$, where $pr[sco_1] \le pr[sco_2] \le \cdots \le pr[sco_N]$ can be obtained as follows: If $p_1 \le p_2 \le \cdots \le p_N$, and there is no violation, then $pr[sco_i] = p_i, i = 1, 2, \ldots, N$. Otherwise, we must go through an iterative pooling process. Let us assume that the current number of elements in sequences $\{p_i\}$ and $\{w_i\}$ is N'(the beginning number is N). If $p_k > p_{k+1}$ for some k (k =1, 2, ..., $N' - 1$), the numbers p_k and p_{k+1} are pooled in the sequence p_1,p_2,...,p'_N and replaced by the resultant single number, which is their weighted average,$\frac{w_k p_k + w_{k+1} p_{k+1}}{w_k + w_{k+1}}$. Likewise,

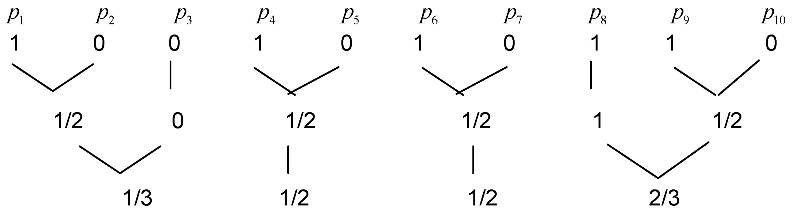
p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 p_{10}
1 0 0 1 0 1 0 1 1 0
1/2 0 1/2 1/2 1 1/2
1/3 1/2 1/2 2/3

Fig. 6. A depiction of the Pool Adjacent Violators Algorithm applied to a small data set. The initial weights of all points are taken to be one.

the numbers w_k and w_{k+1} are replaced in $\{w_i\}$ by the single number $w_k + w_{k+1}$. The result of this operation is an ordered set of $N' - 1$ numbers $\{p_i\}$ and corresponding weights $\{w_i\}$. This procedure is repeated until an ordered set of monotonically nondecreasing numbers is obtained. The effect of pooling order violators is illustrated in the lines of Figure 6. Each line of data represents several pooling operations applied to the data of the previous line. When the algorithm completes, for each i, $pr[sco_i]$ is equal to the particular number within the final set to which the original p_i has contributed.

The PAV algorithm as described here produces a function that is nondecreasing and defined only at the original sco_i values of the data points from which it was derived. To apply the resulting function to a new value, *sco*, $sco_i < sco < sco_{i+1}$, we perform a simple interpolation and define $pr[sco]$ as the average $(pr[sco_i] + pr[sco_{i+1}])/2$. If $sco < sco_1$, we define $pr[sco] = pr[sco_i]$ and likewise at the upper end.

REFERENCES

Adar, E. 2002. A simple and robust abbreviation dictionary. Tech. rep., H. P. Laboratories.

Ayer, M., Brunk, H. D., Ewing, G. M., Reid, W. T., and Silverman, E. 1954. An empirical distribution function for sampling with incomplete information. *Ann. Meth. Statis. 26*, 641–647.

Blum, A. and Mitchell, T. 1998. Comining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory*.

Bowden, P. R., Eventt, L., and Halsted, P. 1998. Automatic arconym acquistion in a knowledge extraction program. In *Proceedings of the ComputTerm98 Conference.* Montreal, Ontario.

Brill, E. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. In *Proceedings of the Computational Linguistics*.

Chang, J. T., Schutze, H., and Altman, R. B. 2006. Creating an online dictionary of abbreviations from MEDLINE. To appear in *JAMIA*.

Charniak, E. 2000. A maximum-entropy-inspired parser. In *Proceedings of the NAACL-2000 Conference*.

Church, K. W. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the 2nd Conference on Applied Natural Language Processing (ACL)*.

Church, K. W. and Gale, W. A. 1991. Probability scoring for spelling correction. *Statis. Comput. 1*, 93–103.

Collins, M. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the ACL Conference*.

Engelson, S. P. and Dagan, I. 1996. Minimizing manual annotation cost in supervised training from corpora. In *Proceedings of the Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing (LNAI) Conference*, S. Wermter et al., eds.

Fauquet, C. M. and Pringle, C. R. 1999. Abbreviations for vertebrate virus species names. *Arch. Virol. 144*, 1865–1880.

FEDERIUK, C. S. 1999. The effect of abbreviations on MEDLINE searching. *Acad. Emerg. Med. 6*, 292–296.

FUKUDA, K., TAMURA, A., TSUNODA, T., AND TAKAGI, T. 1998. Toward information extraction: Identifying protein names from biological articles. *Pac. Symp. Biocomput*, 707–718.

GALE, W., CHURCH, K., AND YAROWSKY, D. 1992. A method for disambiguating word senses in a large corpus. *Comput. Humanities 26*, 415–439.

HARDLE, W. 1991. *Smoothing techniques: With implementation in S. New York*. Spring Verlag, New York.

HATZIVASSILOGLOU, V., DUBOUE. P. A., AND RZHETSKY, A. 2001. Disambiguating proteins, genes, and RNA in text: A machine learning approach. *Bioinformatics 17*, S97–106.

HEARST, M. A. 1991. Noun homograph disambiguation using local context in large text corpora. In *Proceedings of the 7th Annual Conference of the U.W. Centre for the New OED and Text Research*.

HISAMITSU, T. AND NIWA, Y. 1998. Extraction of useful terms from parenthetical experssion by using simple rules and statistical measures. In *Proceedings of the CompuTerm98 Conference*. Montreal, Canada.

HUMPHREYS, B. L. AND LINDBERG, D. A. 1993. The UMLS project: Making the conceptual connection between users and the information they need. *Bull. Med. Libr. Assoc. 81*, 170–177.

JELINEK, F. AND MERCER, R. 1980. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the PWPRP Conference*.

KLAVANS, J., CHODOROW, M., AND WACHOLDER, N. 1990. From dictionary to knowledge base via taxononym. In *Proceedings of the 6th Conference of the UW Contre for the New OED*. Waterloo, Canada.

KOPFF, M., KLEM, J., ZAKRZEWSKA, I., AND STRZELCZYK, M. 1990. Effect of dipyridamole on inosine triphosphate pyrophosphohydrolase activity and inosine triphosphate content in fresh human erythrocytes incubated with adenosine. *Acta. Biochim. Pol. 37*, 227–232.

LANGLEY, P., IBA, W., AND THOMPSON, K. 1992. An analysis of Bayesian classifiers. In *Proceedings of the 10th National Conference on Artificial Intelligence*.

LANGLEY, P. AND SAGE, S. 1994. Induction of selective Bayesian classifiers. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*. Seattle, WA.

LESK, M. 1986. Automatic sense disambiguation: How to tel a pine cone from an ice cream cone. In *Proceedings of the SIGDOC Conference*.

LIDSTONE, G. 1992. Note on the general case of the Bayes-Laplace formula for inductive or a priori probabilities. *Trans. Faculty Actuaries 8*, 182–192.

LIU. H., LUSSIER. Y. A., AND FRIEDMAN, C. 2001. Disambiguating ambiguous biomedical terms in biomedical narrative text: An unsupervised method. *J. Biomed. Inf. 34*, 249–261.

MCCRAY, A. T. 1998. The nature of lexical knowledge. *Methods Inf. Med. 37*, 353–360.

PAKHOMOV, S. 2002. Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical text. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, PA.

PARK, J. C., KIM, H. S., AND KIM, J. J. 2001. Bidirectional incremental parsing for automatic pathway identification with combinatory categorial grammar. *Pac. Symp. Biocomput*, 396–407.

PLATT, J. 1999. *Fast Training of Support Vector Machines Using Sequential Mininal Optimitzation*. MIT Press, Cambridge, MA.

PORTER, M. F. 1980. An algorithm for suffix stripping. *Program 14*, 130–137.

PUSTEJOVSKY, J., CASTANO, J., COCHRAN, B., KOTECKI, M., AND MORRELL, M. 2001. Automatic extraction of acronym-meaning pairs from MEDLINE databases. In *Proceedings of the Medinfo Conference*.

RATNAPARKHI, A. 1998. Maximum entropy models for natural language ambiguity resolution. Ph.D. Dissertation, University of Pennsylvania.

RIMER, M. AND O'CONNELL, M. 1998. BioABACUS: A database of abbreviations and acronyms in biotechnology and computer science. *Bioinformatics 14*, 888–889.

SCHUETZE, H. 1998. Automatic word sense discrimination. *Comput. Linguist. 24*, 97–124.

SCHWARTZ, A. S. AND HEARST, M. A. 2002. A simple algorithm for identifying abbreviation definitions in biomedical text. To appear in *Pac. Symp. Biocomput*.

SINGHAL, A., BUCKLEY, C., AND MITRA, M. 1996. Pivoted document length normalization. In *Research and Development in Information Retrieval.* 21–39.

Stapley, B. J. and Benoit, G. 2000. Biobibliometrics: Information retrieval and visualization from co-occurrences of gene names in MEDLINE abstracts. *Pac. Symp. Biocomput*, 529–540.

Turteltaub, K. W. and Dingley, K. H. 1998. Application of accelerated mass spectrometry (AMS) in DNA adduct quantification and identification. *Toxicol. Lett. 102–103*, 435–439.

Wellner, B. 2005. Weakly supervised learning methods for improving the quality of gene name normalization data. In *Proceedings of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology Conference*.

Wilbur, W. J. 2000. Boosting naïve Bayesian learning on a large subset of MEDLINE. *Proc. AMIA Symp.* 918–922.

Wilbur, W. J. and Kim, W. 2001. Flexible phrase based query handling algorithms. In *Proceedings of the ASIST Annual Meeting*, E. Aversa and C. Manley, eds. Washington, DC.

Yang, Y. and Pedersen, J. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International (ICML'97) Conference*.

Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceeding of the 33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, MA.

Yeates, S., Bainbridge, D., and Witten, I. H. 2000. Using compression to identify acronyms in text. In *Proceedings of the Data Compression Conference*.

Yoshida, M., Fukuda, K., and Takagi, T. 2000. PNAD-CSS: A workbench for constructing a protein name abbreviation dictionary. *Bioinformatics 16*, 169–175.

Yu, H. and Agichtein, E. 2003. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics 19*, Suppl. 1, i340–349.

Yu, H., Hripcsak, G., and Friedman, C. 2002. Mapping abbreviations to full forms in biomedical articles. *J. Am. Med. Inform. Assoc. 9*, 262–272.

Zeitlhuber, U., Haschke, F., Puspok, R., Lechner, K., Knapp, W., and Imbach, P. 1984. Hemophilia and thrombocytopenia in a patient with impaired cellular immunity. A case report. *Blut. 48*, 393–395.

Received December 2002; revised ; accepted August 2006