Dealing with Acronyms in Biomedical Texts

David B. Bracewell, Fuji Ren, and Shingo Kuroiwa *

Abstract

Recently, there has been a growth in the amount of machine readable information pertaining to the biomedical field. With this growth comes a desire to be able to extract information, answer questions, etc. based on the information in the documents. Many of these desired tasks require sophisticated language processing algorithms, such as part-of-speech tagging, parsing, and semantic interpretation. In order to use these algorithms the text must first be cleansed of acronyms, abbreviations, and misspellings. In this paper we look at identifying, expanding, and disambiguating acronyms in biomedical texts. We present an integrated system that combines previously used methods for dealing with acronyms and Natural Language Processing techniques in new way for a new domain. The result is an integrated system that achieves a high precision and recall. We break the task up into three modular steps: Identification, Expansion, and Disambiguation. During identification, each word is examined to determine if it is an acronym or not. For this, a hybrid approach that is composed of a Naive Bayesian classifier and a set of handcrafted rules is used. We are able to achieve results of 99.96% accuracy with a small training set. During the expansion step, a list of possible meanings for the words determined to be acronyms is created. We break the expansion up into two categories, local and global expansion. For local expansion we use windowing and longest common subsequence to generate the possible expansions. Global expansion requires an acronym database to retrieve the possible expansions. The disambiguation step takes the list of possible meanings and determines which meaning is the correct one. To disambiguate the different candidate expansions we use WordNet and semantic similarity. Overall we obtain a recall and precision of over 91%. *Keywords: Acronyms, Text Cleansing, Bioinformatics*

1 Introduction

With the explosion of new information made publicly available from biomedical researchers there has been an equal growth in the amount of acronyms used. Acronyms, abbreviations and misspellings represent a serious problem for Natural Language Processing (NLP) algorithms. Biegert et al. showed that misspellings have negative effects on NLP algorithms [2]. Typically, NLP algorithms make use of lexicons and words not in the lexicon can be thought of as being misspelled. Acronyms and abbreviations that are not common enough to be a part of daily conversation are typically not in the lexicons and as such can be considered as misspelled words, meaning they have negative affects on NLP algorithms. Moreover, the appearance of acronyms and abbreviations in text hinder the automatic creation of the very lexicons that are needed [4]. As such, acronyms, abbreviations and misspellings must be taken care of for NLP algorithms to be useful.

There are many approaches for dealing with misspellings such as [17]. However, for biomedical texts, which are mostly made up of journal articles and conference papers, there should be a low occurrence of misspellings because of the rigorous review process. Because of this the problem of misspellings is likely less important than abbreviations and acronyms.

Acronyms can be thought of as a subset of abbreviations. Abbreviations are shortened forms of a word or phrase. Acronyms are shortened forms of a phrase made up of the initial characters in the words of the phrase. For example, the acronym for “All Nippon Airlines” is “ANA” and the abbreviation for “United States of America” is “USA.” Generally, abbreviations are much more difficult than acronyms, because there is no standard way of shortening a word. For example, “offc.” and “offi.” could both be abbreviations for “office.”

As of late, the line between abbreviations and acronyms is growing thinner and more and more abbreviations are becoming acronym-like. Acronym-like abbreviations are those that are created and function similar to acronyms and have no ending period on them. For example, “DNA” the abbreviation for deoxyribonucleic acid is an acronym-like abbreviation. In biomedical texts, acronyms and acronym-like abbreviations are the most abundant as can be seen in [3]. Because of this, an integrated system for the identification, expansion, and disambiguation of acronyms and acronym-like abbreviations in biomedical texts is proposed. From this point on, acronyms will be defined as acronyms and acronym-like abbreviations. Dealing with these should take care of most problems NLP algorithms would have in dealing with biomedical

*date submitted: 06/14/2006 Department of Information Science and Intelligent Systems The University of Tokushima, JAPAN Email: {davidb,ren,kuroiwa}@is.tokushima-u.ac.jp Author 2 is also at School of Information Engineering, Beijing University of Posts and Telecommunications Beijing 100876, China texts.

The main problem with acronyms is that they are highly polysemous, meaning that one acronym can have meaning expansions. An expansion is the long form of the acronym, for example, “All Nippon Airlines” is an expansion for “ANA.” In general, this property of having many expansions can be dealt with, because the expansions can be partitioned into much smaller domain sets where all the members of a set belong to a common domain. If the domain of the text is known then the possible expansions for the acronym can be reduced. For example according to Acronym Finder [1] the acronym “TWA” has 20 possible expansions, but depending on the choice of domains these 20 expansions can be partitioned into nine domains with the largest domain containing only five expansions. Dealing with five possible expansions is much more desirable than dealing with 20 possible expansions. The list below shows some of the expansions of “TWA.”

- Trans Word Airlines
- Transcontinental and Western Airlines
- Tail Wire Antenna
- Thin-Wire Antenna
- Trailing Wire Antenna
- Teen Wrestling Association
- Texas Wrestling Association
- Texas Wrestling Alliance

Acronyms in non-biomedical texts, henceforth referred to as the general case, exhibit this ability to break down expansions into smaller domains. Many approaches for the general case such as [20], [11], [16] have achieved good precision. However, the acronyms that are inside of the biomedical domain do not exhibit this ability and are still highly polysemous as [8] shows. General acronym expansion techniques perform poorly on medical texts as [13] shows. As such, approaches for their expansion, as of yet, have not caught up with the results for the general case. Therefore, an integrated system that combines previously used methods for dealing with acronyms and Natural Language Processing techniques in new way for biomedical texts is proposed. This paper makes use of a Naive Bayesian classifier for identification of the acronyms. Then it uses a windowing technique and the longest common subsequence to find candidate expansions. If no candidates can be found using this method an acronym database is consulted. Finally, for disambiguation a conceptual clustering algorithm is uses semantic similarity.

The system proposed in this paper has three parts; Identification, Expansion, and Disambiguation. Identification is the process of determining which words in a text are acronyms. Expansion deals with finding the possible long forms (expansions) for the identified acronyms. Disambiguation determines which long form is the correct one for the acronym in the given text. This process is summarized below.

1. Identify acronyms in the text
2. Find candidate expansions for the acronyms
3. Disambiguate to find the correct expansion.

The paper will proceed as follows. First, the identification method that is used will be examined. Next, the expansion process will be discussed. The expansions will be categorized into two different types of expansions and methods will be proposed for dealing with each. Then, disambiguation and how to find semantic similarly between expansions and texts will be shown. Next, the training and testing data used will be looked at. Then, experimental results will be shown and discussed. Next, related work in the field will be examined. Finally, the paper will end with future work and concluding remarks.

2 Identification

Identifying acronyms is the first the step in dealing with them. Other approaches to acronym expansion, dealing mostly with the general case, includes Park and Byrd’s approach that uses a set of conditions and rules for identification [11]. In addition, Taghva and Gilbreth used a simple assumption that acronyms are words that have a length between 3 and 10 characters and are all capital letters [16].

This paper takes a hybrid approach composed of a Naive Bayesian classifier, using a maximum a posterior approach, and two handcrafted rules. The Naive Bayesian classifier is a conditional model that assigns the most likely class to a set of independent features. Equation 1 shows the calculation used to determine a class given a set of features. Even though its assumption that the features (variables) are independent it has been shown to generally do well in classification [15].

$$P(C|F_1,\ldots,F_n) = P(C)\prod_{i=1}^{n} P(F_i|C) \tag{1}$$

In this paper, five features were looked at for the identification process and are shown below.

1. In Parentheses: Boolean value indicating if the word appears in a parenthesis.

[1] http://www.acronymfinder.com/

2. In Dictionary: Boolean value indicating if the word is in the dictionary.
3. Capital Letter Percentage: Percentage of letters that are uppercase (rounded to the nearest percent).
4. Consonant Letter Percentage: Percentage of letters that are consonants (rounded to the nearest percent).
5. Length: Length of the word.

The Bayesian classifier used the first four features and the handcrafted rules used the last one. The "In Parentheses" feature is a boolean value indicating if the word being looked at is in parentheses. The "In Dictionary" feature is a boolean value indicating if the word being looked at is in the dictionary. The "Capital Letter Percentage" and "Consonant Letter Percentage" features represent the percentage of characters that are capitalized and are consonants respectively. However, during testing it was found that using a threshold to convert the values into a binary value increased the performance. The threshold was chosen to be 66% for both. These values were chosen to maximize performance for acronyms with length greater than 3 characters and because it caused the highest performance increase during testing. If 50% or less was chosen then there would have been problems with 2 letter words being misidentified when they appear as the first word in the sentence. The increasaed performance by the thresholding is probably due to simplfying the search space. Modelling probabilities with binary values instead of integer values ranging from 0 to 100 should give a more accurate probability estimate when the amount of training data is small.

Two commonsense handcrafted rules were used after the classifier. The rules can be seen below.

1. IF all lower case AND in parenthesis THEN not acronym
2. IF all upper case AND length > 1 THEN acronym

The first rule fixes overfitting caused during the training process. The second handcrafted rule takes care of the general acronym case. This rule is similar to the one Taghva and Gilberth used only less restrictive [16].

In comparison, Ao and Takagi [1], Park and Byrd [11], and Taghva and Gilberth [16] used only handcrafted rules. Rule-based identification using handcrafted rules seems to be the typical way of identifying acronyms and typically gives good results. These rule based approaches often use the length of the word, the number of capital letters, the number of vowels, if the word contains numbers, consonant patterns and if the word is in parentheses as features [1], [11], [16]. Other research like [8] and [19] ignore the problem of identifying acronyms completely.

3 Expansion

Expansion can be broken down into two categories: global and local. Global expansion means the expansion is not given in the text. Instead the acronym is considered to be common to the domain and readers are expected to already know it. The only way to deal with these types of expansions is to use an external acronym database. This paper has chosen to use the acronym database from UMLS (Unified Medical Language System)[2]. UMLS is a project sponsored by the United States National Library of Medicine and was created to aid in the processing of biomedical texts using NLP. It currently contains over 17,000 acronyms. It was used, because it is freely available and the data is based on the text in PubMed[3], which is used for the experimentation. While there are many freely available acronym lists, UMLS is the biggest and most comprehensive for the biomedical domain. Other lists like the Biomedical Acronym Resolver[4] and Acromed[5] contain more acronyms, but the lists are automatically created and contain errors in them. In addition, these lists usually only allow online access and the underlying databases cannot be downloaded.

Local expansion means that somewhere in the text, typically to the left or right of the first occurrence, an expansion is given for the acronym. Using this knowledge windows can be constructed around the different occurrences of the acronym. The text in the windows then becomes the candidate expansions. A window is simply a consecutive sequence of words. The number of windows created is proportionate to the number of times the acronym occurs in the text. Each acronym occurrence will have two windows, a left and a right, associated with it. So, for example, if an acronym occurs five times there will be ten windows associated with the acronym. The window size is determined by the length of the acronym in terms of characters. The size of a window is equal to the number of words in the window. Stop words (and, the, of, etc.) and numbers are ignored when creating the window and do not count in determining the size of the window. The stop word list was created by us and only includes English prepositions and conjunctions.

In addition to local and global expansion, plural and variant forms of acronyms must also be addressed. A variant form of an acronym is typically made up of a base and number. For example, the acronym TT could be defined in a paper as time trial and then later on TT1 and TT2' could be seen. It is obvious to the reader that TT1 and TT2's expansions are time trial 1 and time trial 2. These

[2] http://www.nlm.nih.gov/research/umls/
[3] http://www.ncbi.nlm.nih.gov/PubMed
[4] http://invention.swmed.edu/argh/
[5] http://medstract.med.tufts.edu/acro1.1/index.htm

variant forms can be thought of as local expansion, but in order to get their expansion the base form's expansion has to be looked at. To group plural and variant forms of acronyms with their base forms two rules are used. The first rule checks if there is a lowercase 's' at the end of the acronym. If there is, we see if there is another acronym that is exactly the same except for the lowercase 's.' The second rule is similar to the first one except we check for digits instead of a lower case 's.'

The expansion method employed makes use of the one sense per discourse rule introduced by [5] and used by [19] in the domain of acronyms. One sense per discourse in terms of acronyms means that all occurrences of an acronym, in an abstract, have the same expansion. Yu, Tsuruoka, and Tsujii find expansions for each acronym occurrence then choose one by majority voting [19]. Unlike them we combined the information from all of the occurrences to find the expansion. The expansion method used in this paper tries to exploit local expansion as much as possible. The overall expansion process is done in the following four steps:

1. Group acronyms
2. Local expansion with restrictive longest common subsequence
3. Global expansion
4. Local expansion with non-restrictive longest common subsequence

The grouping acronyms step deals with grouping the plural and variant forms with their base form. Next, the algorithm attempts to extract local expansion candidates using the windowing technique and the longest common subsequence (LCS) and is based on [16]. Each of the words in a window has its first letter extracted and the LCS is performed on the resulting string and the letters in the acronym. Each window is given a score that is the percentage of the acronym that the LCS covers. For example, if the window contained "body mass index" then after extracting the first letters the resulting string would be "BMI." If the acronym being looked at was "BMI" then the resulting score for the window would be 100. The window with the maximum score is returned. To eliminate erroneous candidate windows the score must be above 50. If this step fails to yield an expansion then global expansion is attempted using the acronym database. If for some reason the acronym database does not contain the acronym then another less restrictive local expansion is done. Less restrictive means that all the letters from the windows are used when doing the LCS and every window that has an LCS length equal to the length of the acronym will become a candidate expansion.

4 Disambiguation

After candidate expansions are gathered for an acronym the correct expansion must be chosen. This is the job of the disambiguation module. Some of the methods employed for disambiguation of acronyms in biomedical texts are Support Vector Machines [19] and Maximum Entropy [10]. These approaches have two main problems. The first being that they are solely reliant on acronym databases. The second is that they completely ignore the benefits of local expansion since they are reliant on acronym databases. As such, this paper uses a new approach. Conceptual clustering based on semantic similarity is used. Each cluster describes a set of semantically similar senses computed using WordNet [9].

WordNet is a lexical dictionary that organizes words in a hierarchical structure based on psychological principals [9]. The individual words are organized in synonym sets and have attributes like "is a" and "member of." Semantic similarity is way of telling how similar two words are based on their semantic meaning. For example, "house" and "home" are semantically similar, because they both describe dwellings. Because WordNet organizes words in groups of synonyms and in a hierarchical structure based on concept it is useful for computing semantic similarity.

In this paper, the semantic similarity method proposed by Jiang and Conrath [6] was used. This method uses information content and the edge distance between the sense nodes in WordNet. Information content uses the "is a" hierarchy of WordNet to determine how much information two words have in common [14]. This method was chosen for its high correlation to human judgment and its superior performance to the standard edge counting and information based counterparts.

First, the text is described in a set of sense clusters. To do this the top 15% of the words occuring in the abstract, omitting stop words, numbers, and acronyms, are used. These words are looked up in Wordnet and their noun senses are extracted. The noun senses are then clustered using semantic similarity. Figure 1 shows an overview of the clustering process. The reason for a restriction of 15% is solely a performance issue. Semantic similarity is slow and the 15% gave a good blanance of speed and performance.

There are no initial clusters and new clusters are created when a noun sense is not similar to any other cluster. As such, the first noun sense that is examined will create a new cluster. To compute the similarity between a sense (S) and a cluster (C) we simply find the maximum similarity score between S and each sense in C, see equation 2. Assuming the similarity measure is accurate then this equation should work well.

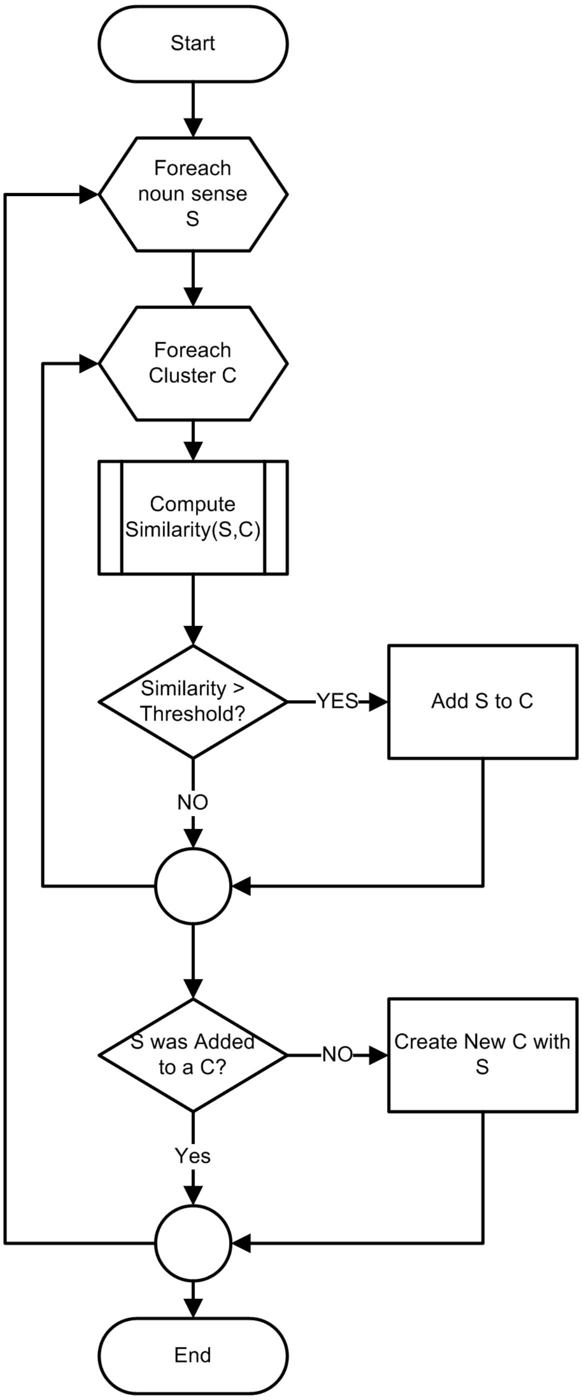

Figure 1: Clustering Process

$$Similarity(S,C) = \underset{cs_i \in C}{\arg\max}(Similarity(S,cs_i)) \quad (2)$$

$$Score(E,T) = \sum_{C_i \in T} Similarity(E,C_i) \quad (3)$$

$$Similarity(E,C) = \mid C \mid \times \sum_{S_i \in E} Similarity(S_i,C) \quad (4)$$

No attempt is made to disambiguate word senses in the text, instead all word senses are used. The ideal is that the correct senses will belong in larger clusters, thus doing a sort of self disambiguation. After the clusters are created for the text each expansion candidate is looked at individually. Each expansion (E) is assigned a score based on the similarity between it and the text (T) as shown in equation 3. The similarity measure for an expansion and a cluster is simply the sum of the similarity measures for each sense of each word in the expansion and the cluster, see equation 4. The candidate expansion with the highest score is then chosen as the correct expansion. The reason that the correct expansion will receive the highest score is that, hopefully, the clusters that accurately describe the text will be much larger than those that do not and as such if the words in the expansion are more similar to those in the larger clusters then the expansion should be closely related to the text.

5 Training and Testing Data

For training and testing data, abstracts extracted from PubMed[6] were used. PubMed currently contains over 15 million citations from various journals. 300 randomly extracted abstracts of biomedical data were used for the experimental data. This set was then split up into 20 abstracts for training and the rest for testing. The 300 abstracts had over 56,000 words and resulted in 562 unique acronyms with a total of 1,728 occurrences, 61.25% of which had local expansions.

The abstracts have in them section headings. These section headings are in all capital letters and cause problems for identification purposes. As such an ignore list which is made up of these section headings was created. Other data files that we used were a dictionary[7] and a list of stop words. The dictionary is used during identification for the *InDictionary* feature and the list of stop words is used during expansion.

[6] http://www.ncbi.nlm.nih.gov/PubMed

[7] http://www.dcs.shef.ac.uk/research/ilash/Moby/

6 Results

6.1 Identification

For identification the Naive Bayesian Classifier is compared to the approaches of [11] and [16] using recall, precision, F-Measure, and accuracy to compare the results. Recall is a measure of how well the method was able find all the "real" acronyms, see equation 5. Precision is a measure of how accurate the method is in identifying only "real" acronyms as acronyms, see equation 6. F-Measure is a way to combine recall and precision into one measure [18], see equation 7. Accuracy is a measure of how well the method is at identifying words (acronyms and non-acronyms,) see equation 8.

$$Recall = \frac{\text{\# of Correctly Identified Acronyms}}{\text{\# of Acronyms In Abstract}} \tag{5}$$

$$Precision = \frac{\text{\# of Correctly Identified Acronyms}}{\text{\# of Acronyms Found}} \tag{6}$$

$$F - Measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{7}$$

$$Accuracy = \frac{\text{\# of Correctly Identified Words}}{\text{Total \# of Words}} \tag{8}$$

20 randomly chosen abstracts were used for training and testing was done on the remaining 280. This was repeated two more times, each time picking a new random set of 20 training abstracts. In table 1, we show the average results of the methods over the three different testing sets. NB is the Naive Bayesian classifier without the handcrafted rules and Hybrid is the Naive Bayesian classifier with the handcrafted rules. Park and Taghva refer to the approaches by [11] and [16] respectively.

Typically, there is a trade off in precision and recall, meaning when precision is high the recall is low and vice versa. That trade off can be seen in the approaches by Park and Byrd and Taghva and Gilbreth. Taghva and Gilbreth's approach has a low recall, but the precision is high. This is due to the fact that their assumption, that an acronym is a word three to ten characters long and all uppercase, is too restrictive to find all the acronyms, for example it could not find "MtDNA" as an acronym. However, because it is so restrictive the words it does identify as acronyms are usually acronyms. Park and Byrd's approach has a much higher recall rate, but the precision is extremely low. There assumptions are much less restrictive and are designed to identify acronyms in the general case. Because their method is not so restrictive they can find most of the acronyms, but they also identify many non-acronyms as acronyms. Using just the Bayesian classifier resulted in a recall and precision of greater than 91%. The simple handcrafted rules were able to help boost the precision, recall, and accuracy. In this case both the Bayesian and Hybrid methods are able to overcome the standard trade off that is seen between recall and precision.

The hybrid clearly outperforms the other methods. The Bayesian classifier was able to learn which attributes were most closely related to acronyms and non-acronyms. The addition of the hand-crafted rules helped in correcting overfitting. The only drawback for the proposed approach is that it requires a small training set. In this paper, 20 abstracts were used as training data. However, the performance gain over the previous methods, we believe, justifies the effort involved in training.

6.2 Expansion and Disambiguation

For testing expansion the recall rate was used as a comparison. Recall rate, for expansion, means the number of acronyms that had expansions found divided by the total number of acronyms, see equation 9. The results were broken down further into the type of expansion.

$$Recall = \frac{\text{\# of Acronyms with expansions}}{\text{Total \# of Acronyms}} \tag{9}$$

For testing disambiguation, precision is used as a comparison. The similarity measure by [6], discussed earlier was used. The threshold was 0.8, which was chosen through experimentation.

Table 2: Expansion & Disambiguation Results

Expansion Type	**Recall**	**Precision**
Local	97.25%	95.14%
Global	81.70%	80.98%
Local & Global	93.21%	91.93%

In table 2 results for the entire set of 300 abstracts can be seen. It can be seen that the local expansion method performs quite well in recall (expansion) and precision (disambiguation.) The local expansions tended to have only a few candidate solutions and as such their precision is much higher. The global expansions tend to have many possible expansions, which makes disambiguation more complicated.

The results for global expansion were not as good. Both recall and precision were lower than that of local expansion. Since recall rate for global expansion is directly dependent on the size of the underlying acronym database it can be concluded that the acronym database is inadequate. The inadequate database also adversely affects

Table 1: Identification Results

Method	Recall	Precision	F-Measure	Accuracy
NB	92.35% (±.8%)	97.23% (±.7%)	94.73% (±.7%)	99.73% (±.0%)
Hybrid	99.22% (±.0%)	99.45% (±.0%)	99.34% (±.0%)	99.96% (±.0%)
Park	95.93% (±.2%)	39.71% (±.2%)	56.17% (±.2%)	96.04% (±.1%)
Taghva	66.75% (±.3%)	99.56% (±.1%)	79.92% (±.2%)	99.11% (±.0%)

the precision. In addition the coverage of WordNet may also hinder the results. It was found that many of the more complicated biomedical terms were not present in WordNet. These obstacles can be overcome in the future by mining abstracts for acronyms and expansions using the restrictive local expansion technique in order to help increase the size of the acronym database. The draw back to this is that many new possible expansions will be introduced, which may cause an overall decline in precision. To help in the precision a biomedical ontology must be integrated with the WordNet ontology.

The overall recall was just under 94%. In contrast, Pustejovsky et al. reported in [13], that Acrophile [7] had a recall of 60% and Acromed [13] had a recall of 72% on similar texts. The overall precision was just under 92%. This shows that these methods, while having room for improvement, are quite good and the balance of recall and precision is better than other systems.

Finally, in table 3 the overall results of the system can be seen. Like testing the identification method a three-way-cross-validated experiment was done. The results show that the system achieves better than 91% for recall, precision, and F-Measure.

Table 3: Overall System Results

Measure	Result
Recall	91.90% (±.44%)
precision	91.29% (±.42%)
F-Measure	91.60% (±.42%)

6.3 Error Analysis

There were two types of errors; No expansion and Incorrect expansion. All acronyms marked as having local expansion had the “no expansion” error. This means that the local expansion technique could not find any candidates and that the acronym was not in the acronym dictionary. For the acronyms marked as having global expansion, roughly 70% had “no expansion errors” and the other 30% were “incorrect expansion” errors. Table 4 shows some example errors. In the table it can be seen that the “incorrect expansion” errors were often caused by the acronym not being specific to the biomedical domain. These acronyms had expansions in the biomedical domain, but the correct expansion was either not in the dictionary or another expansion was more semantically similar. In all, most errors were of the “no expansion” type. To help reduce these errors are more comprehensive acronym dictionary and less restrictive local expansion methods should be investigated.

7 Related Work

Little work has been done in creating a complete system for the identification, expansion and disambiguation of acronyms in biomedical texts. One effort to do so is ALICE [1]. While ALICE claims to achieve a recall of 98% and a precision of 96% the authors made one crucial error. The error is that for identification they only created rules that looked inside of parentheses for acronyms or expansions. Thus, the only acronyms they will be able to find are those that are in parentheses or their expansions are in parentheses. In our experimental data, acronyms that were in parentheses made up only about 19.5% of the total. We did not collect data on the expansions, but with over 38% of the acronyms having global expansions it is not possible for such a technique to achieve such a high recall on our experimental data.

Most of the research in the field has been focused on algorithms for building acronym databases. When building an acronym database precision is of key importance and a lower recall can be tolerated. Acromed [12] is one such system and was able to achieve a precision of 98% with a recall of 72% [12]. Acromed and other algorithms work well for this task. However, with a low recall they are not as useful for a system that acts a preprocessor for language processing algorithms. The goal of such a system is to expand as many acronyms as possible as precisely as possible, i.e. balance recall and precision.

Another area of research in the field has been in dealing with the disambiguation process. Maximum Entropy [10] and Support Vector Machines [19] are two such methods employed for disambiguation. These methods use an acronym database/list to find the possible expansions. Then using context information for the expansions and the acronym they match the expansion to the acronym. One problem with these techniques is that they completely ignore the benefits of local expansion, which can greatly improve the accuracy by limiting the possible expansions. In addition they are completely dependent on the acronym database/list they use. Their systems' performance is limited by the acronym database they have. For example, Pakhomov used the acronym database from UMLS like we did and, we found that, this database is

Table 4: Expansion Errors

Acronym	Frequency	Type	Correct	Suggested
SD	3	Global	Standard Deviation	Skin Destruction
CI	23	Global	Confidence Interval	Cochlear Implant
MA	1	Global	Maryland	Meter Angle
QEEG	1	Local	Quantitative EEG	NONE
QTD	3	Local	QT-dispersion	NONE
MEDASP	2	Local	medical ASP	NONE

not comprehensive enough. These systems could be modified to take advantage of local expansion, but they still have a problem with the acronym list.

As far as we know, this paper is one of the few that proposes a complete system for identifying, expanding, and disambiguating acronyms in the biomedical domain. While there are many web sites that allow a user to search for an acronym and return a list of expansions those systems do not tell the user which expansion is the correct one in terms of the article they are reading. The intention of this system is to automatically replace acronyms with their long forms in texts so that NLP algorithms can achieve better performance.

8 Conclusions

In this paper we have presented a new modular approach for identifying, expanding, and disambiguating acronyms in biomedical texts. There exists many good methods for dealing with the general case of acronyms, but these methods unfortunately do not work well on the biomedical domain. Previous research dealing with the biomedical domain has mainly been focused on building acronym databases or just disambiguating the acronyms. But, as we showed in this paper typical identification methods were not so reliable when using them on the biomedical domain.

For identification we used a Naive Bayesian classifier with two handcrafted rules. The classifier was able to achieve a recall and precision of greater than 99% with a small training set. For finding expansions we tried to exploit the idea of local expansion as much as possible and achieved a recall of over 97% for them. Finally, for disambiguation we presented a method that clusters words from an abstract based on similarity. We then used these clusters to help find the correct expansion for an acronym.

While the presented system was designed for the biomedical domain, it should be applicable to other domains. New training data for the Bayesian classifier would be required as well as an acronym database. Both should be easily obtainable as there are numerous acronym databases online and many free sources of text that have acronyms.

In the future we hope to achieve better results by improving the acronym database and coalescing a biomedical ontology into WordNet. We also would like to test the identification method on the general case and see how it compares to the standard methods presented. Finally, we would like to explore different similarity measures in the clustering phase to see if better results can be obtained. The similarity measure is important for different reasons. One reason is that computing semantic similarity is the most intensive part of the system and as such the running time is determined by the similarity measure. The second reason is that it has a direct relation to the quality of clusters and in choosing the correct expansions.

Acknowledgment

This research has been partially supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan under Grant-in-Aid for Scientific Research (B), 14380166, 17300065, Exploratory Research, 17656128, 2005. We would also like to thank the reviewers for their valuable comments.

References

[1] H. Ao and T. Takagi. An algorithm to identify abbreviations from medline. *Genome Informatics*, 14:697–698, 2003.

[2] J. Bigert, O. Knutsson, and J. Sjobergh. Automatic evaluation of robustness and degradation in tagging and parsing. In *In Proceedings of the 2003 International Conference on Recent Advances inNatural Language Processing*, pages 51–62, 2003.

[3] J. T. Chang, H. Schutze, and R. Altman. Creating an online dictionary of abbreviations from medline. *The Journal of the American Medical Informatics Association*, 9:612–620, 2002.

[4] C. Friedman, H. Liu, L. Shagina, S. Johnson, and G. Hripcsack. Evaluating the umls as a source of lexical knowledge for medical languageprocessing. In *Proceedings of American Medical Informatics Association Symposium*, pages 189–193, 2001.

[5] W. Gale, K. Church, and D. Yarowsky. One sense per discourse. In *Proceedings of the DARPA Workshop on Speech and Natural Language Processing*, pages 233–237, 1992.

[6] J.J. Jiang and D.W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the Int'l Conf. on Research on Computational Linguistics*, 1997.

[7] L. Larkey, P. Ogilvie, M. Andrew Price, and B. Tamilio. Acrophile: an automated acronym extractor and server. In *Proceedings of the fifth ACM conference on Digital libraries*, 2000.

[8] H. Liu, AR. Aronson, and C. Friedman. A study of abbreviations in medline abstracts. In *Proceedings of American Medical Informatics Association Symposium*, pages 464–469, 2002.

[9] G. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38:39–41, 1995.

[10] S. Pakhomov. Semi-supervised maximum entropy based approach to acronym and abbreviationnormalization in medical texts. In *Proceediags of the 40th Annual Meeting of the Association for ComputationalLinguistics (ACL)*, pages 160–167, 2002.

[11] Y. Park and R. Byrd. Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of the 2001 Conference on Empirical Methods in Natural LanguageProcessing*, pages 126–133, 2001.

[12] J. Pustejovsky, J. Castano, B. Cochran, M. Kotecki, and M. Morrelland A. Rumshisky. Extraction and disambiguation of acronym-meaning pairs in medline. In *Proceedings of Medinfo*, 2001.

[13] J. Pustejovsky, J. Castano, B. Cochran, M. Kotecki, and M. Morrelland A. Rumshisky. Linguistic knowledge extraction from medline: Automatic construction ofan acronym database. In *Proceedings of Medinfo*, 2001.

[14] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995.

[15] I. Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 2001.

[16] K. Taghva and J. Gilbreth. Recognizing acronyms and their definitions. Technical report 95-03, ISRI (Information Science Research Institute) UNLV, 1995.

[17] Sebastian Van Delden, David B. Bracewell, and Fernando Gomez. Supervised and unsupervised automatic spelling correction. In *Proceedings of the 2004 IEEE International Conference on Information Reuseand Integration*, pages 530–535, Las Vegas, NV, November 2004.

[18] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Researchand development in information retrieval*, pages 42–49, 1999.

[19] Z. Yu, Y. Tsuruoka, and J. Tsujii. Automatic resolution of ambiguous abbreviations in biomedical texts using support vector machines and one sense per discourse hypothesis. In *Proceedings of the SIGIR'03 Workshop on Text Analysis and Search for Bioinformatics*, pages 57–62, 2003.

[20] M. Zahariev. Efficient acronym-expansion matching for automatic acronym acquisition. In *Proceedings of the International Conference on Information and KnowledgeEngineering*, pages 32–37, 2003.