MEDINFO 2001
V. Patel et al. (Eds)
Amsterdam: IOS Press

Automatic Extraction of Acronym-meaning Pairs from MEDLINE Databases

James Pustejovsky[a], José Castaño[a], Brent Cochran[b], Maciej Kotecki[b], Michael Morrell[a]

[a] *Laboratory for Linguistics and Computation at Brandeis University, Waltham, MA*
[b] *Department of Physiology at Tufts University, Boston, MA*

Abstract

Acronyms are widely used in biomedical and other technical texts. Understanding their meaning constitutes an important problem in the automatic extraction and mining of information from text. Here we present a system called ACROMED that is part of a set of Information Extraction tools designed for processing and extracting information from abstracts in the Medline database. In this paper, we present the results of two strategies for finding the long forms for acronyms in biomedical texts. These strategies differ from previous automated acronym extraction methods by being tuned to the complex phrase structures of the biomedical lexicon and by incorporating shallow parsing of the text into the acronym recognition algorithm. The performance of our system was tested with several data sets obtaining a performance of 72 % recall with 97 % precision. These results are found to be better for biomedical texts than the performance of other acronym extraction systems designed for unrestricted text.

Keywords

Medical Informatics, Information Storage and Retrieval; Pattern recognition, Abstracting and Indexing, Acronyms, Medline.

Introduction

The use of computational techniques to automatically extract information from biomedical databases and in particular from MEDLINE abstracts has received particular attention recently (e.g.:[1][2][3][5]). The lexical database for acronyms and abbreviations in UMLS, however, contains only about 10,410 entries. This is a very small portion of the acronyms occurring in the Medline database. The problem of determining the meaning of acronyms in text is related to that of alias determination and entity identification. It is necessary to know to what entity an expression refers in the text in order to accurately extract and categorize the interactions and relations between terms as expressed in the text.

The problem of determining automatically the meaning of acronyms in biomedical texts is both a critical one as well as a difficult one. It is critical because the performance of information retrieval and extraction tasks is significantly degraded when acronym meanings are not properly understood or interpreted. The problem is exacerbated in the biomedical literature by the widespread use and frequent coinage of acronyms. The problem is difficult because there is wide variance in conventions within the biomedical communities on forming acronyms from their "long forms". In the past few years a number of interesting techniques have appeared that determine automatically the meaning of an acronym in free text [4][6][7][8]. The best results from those techniques are summarized in Table I below.

Table 1. Precision and Recall from free text

Acronym Extraction Systems	Precision	Recall
[4]Canonical/ Contextual	87%	88%
[4]Canonical	96%	60%
[4]Canonical Simple	94%	59%
[7]Pattern matching	68%	91%
[8]Data Compression	90%	80%

Although some of these results are good, they are far from optimal and we have found their performance is significantly degraded when applied to biomedical texts. Since our goal is to automatically populate databases and supply accurate information about these entities much higher precision is required. In this paper, we present the results of two experiments we performed to derive the long form meaning of acronyms in Medline abstracts.

The first experiment implemented a pattern-matching algorithm that identifies an acronym, and then moves left in the input string to determine candidates for the long form of the acronym.

In the second experiment, we constrain and circumscribe the application of a pattern-matcher after having performed a robust phrase-level parsing of the input string. Once the proper syntactic structure is assigned to the Noun Phrase within which a potential acronym may occur, we apply a finite-state matching algorithm with considerable precision to identify the long form. Both the precision and recall of this technique are significantly greater than that achieved in previous work. The reason for this marked improvement is

due to several factors. Conventional approaches to acronyms have conflated two computationally distinct problems:

a) Determining the window size of the text within which the long form for the acronym lies.

b) Identifying the long form by matching, deleting and simplifying character strings relative to the acronym itself.

We show that much greater accuracy can be attained if these two problems are treated as separate computational tasks. Importantly, the first problem is solved by a constrained context-free parsing algorithm, developed independently for the automated interpretation and extraction of protein and gene descriptions and their relationships in biomedical text in our larger project called Medstract (www.medstract.org).

Materials and Methods

General Design of the Experiments

The identification of acronym-meaning pairs is embedded in the (simplified) architecture of an engine to extract imformation from Medline abstracts seen in Figure 1.

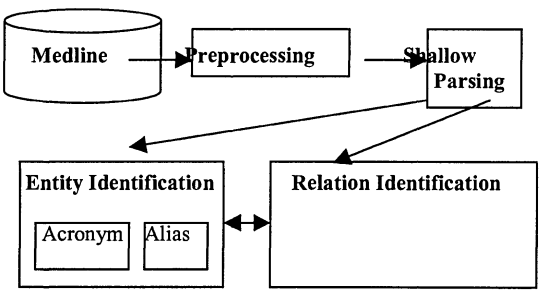

Figure 1 General Architecture

Medline abstract XML files are preprocessed first (Medline field and tag identification, tokenization, stemming), followed by the syntactic machinery performing syntactic tagging and shallow parsing on the body of the abstracts. The modules of Entity Identification and Relation Identification have their own sub-modules. The acronym-meaning extraction module is embedded in the Entity identification module, which contains similar and interacting modules, such as alias identification, anaphora resolution or semantic type identification. We consider the acronym-meaning identification to be a subclass of the alias identification problem, understanding that acronyms are a subtype of aliases. Here we will not address the issue of what constitutes an acronym, given that the boundaries between acronyms, abbreviations and alias can be quite fuzzy.

The goal of the task presented in this paper is to capture the meaning of those acronyms which are introduced in the text as standing for its long form. Acronyms assumed to be known by the reader are not the target of the task to be performed: e.g., if the acronym "HIV" occurs in an abstract without its long form ("Human Immunodecifiency Virus") in its proximal context, it is not a target for our procedure. It is assumed in this case that the knowledge of what the entity "HIV" stands for must be found elsewhere: a lexical database or another abstracts.

The peculiarities of entity names in the Medline Database makes this task particularly difficult: number and symbol combinations, use of compound words, dashed words, as can be seen in the examples below, makes the problem quite hard:

SOD1: Cu/Zn superoxide dismutase
F1 + 2: prethrombin F1 + 2 fragment
E2: estradiol-17 beta
CaSR: Ca2+-sensing receptor

In evaluating our procedures, we followed the general standards for corpus preparation and experimental design now established in the Information Retrieval community. A set of (86) Medline abstracts from 1997-8 was randomly selected using a search engine. This set of abstracts was manually annotated for all occurrences of alias pairs by a biomedical specialist. It contains 155 occurrences of Acronym-meaning pairs. This was established as the Gold Standard for our development corpus.[1] We used the development corpus to test and improve our set of regular expression strategies.

Another set of 100 abstracts was randomly selected from the results of a search for the term "*gene*" in abstracts from a diverse, but small group of high impact biomedical journals. This set of abstracts contains principally molecular biology abstracts whereas the development corpus has more medical and clinically related abstracts in addition to molecular biology abstracts. These abstracts were manually annotated used as the Gold Standard Evaluation Corpus. It contains 173 occurrences of alias pairs

The original Gold Standards were annotated as Gold Standards for the Alias-of relation (as mentioned above). Examples of the Alias-of relations as marked by the domain experts are as follows:

> "**ASR**" (an acronym) can stand for "automatic speech recognition";
> "**Spherix**" for "Bacillus sphaericus B-101, Serotype H5a,5b" ;
> "**rhG-CSF**" for "filgrastim" and
> "**PaO2**" for "partial pressure of oxygen in arterial blood".

[1] The Gold Standards can be checked at http://www.medstract.org using a browser compatible with XML documents.

The last 3 of these examples are clearly aliases that are not acronyms, whereas ASR is a straightforward acronym. Identification of the alias pairs revealed that only 6 of the 165 alias pairs in the sample were clearly not acronyms. Thus, this analysis lead us to an important (though preliminary) conclusion that the acronym-meaning pair is the most important in the Alias-of relations.

Given that acronyms were the predominant expression of the aliasing relation in the biomedical domain, we decided to focus on the acronym meaning problem. Thus, from the 165 pairs marked in the development Gold Standard, 6 alias pairs which clearly were not acronyms were removed,[2] while several others were left in the set which could be considered marginal cases. The cases that were removed are more complex alias expressions than acronym-meaning pairs and should be handled by a different strategy. Such a strategy would need to utilize information outside of simple strings and thus were eliminated from the Gold Standard.

The implementation

Our strategy for extracting Acronym-meaning (meaning: also called "long form" or "expansion") in the Medline database, was developed following two different tracks.

First we considered the problem of recognizing acronym-meaning pairs as the problem of finding two strings in a text that satisfy certain properties: match specific regular expressions. We will call this the *regular expression algorithm*. The input text is a simple sequence of strings. This is basically the same strategy for this problem that was used by the works mentioned in the previous section. We designed regular expressions matching a potential acronym and looked for its meaning in the context. Some subroutines convert the potential acronym into a regular expression. This regular expression is used to search in the close context from the position where the potential acronym was found. When a string that matches the potential acronym is found, it is rated with a formula that compares how good the acronym is to a comparison or threshold measure. We implemented a very restricted pattern for the acronym-meaning pair ("#" stands for a sentence boundary):

$String_i$ "(" $String_j$ ")".

Then each of its composing characters are attempted to match as a prefix or infix of the words that compose $String_i$. If there is a match (a suffix of $String_i$ that starts with the same character/symbol in the acronym) it is assigned a score according to the formula (see [6] for a similar approach):

$$\text{Score} = \frac{\text{\# of words in the match}}{\text{\# of characters in the acronym}}$$

[2] From the Evaluation Gold Standard five pairs were removed.

If the score is below some threshold (our best results were with threshold 1.5), then the pair is accepted. The pattern we used was quite limited and might correspond to a subset of the Canonical Simple pattern in [1]. We decided to address the simplest canonical form because it was the most frequent and constrained case. We understood a more general pattern would be more prone to errors. An important issue not addressed by the algorithm used here, but which would be expected to improve the recall, would be to add the capability of looking to the right side context in addition to the left side.

The second approach we evaluated was a refinement of the previous one. Although the basic problem remains the same: two strings are compared to decide if one is an expansion or meaning of an acronym, the extent and boundaries of the context where this expansion is searched for and solved is totally different. This Acronym-meaning extracting machine uses the machinery that we have developed to extract information from the Medline database - specifically pre-processed text annotated with syntactic information. The input to this algorithm is not just raw text (sequences of strings), but a shallow parsed text. Shallow parsing is a technique widely used in information retrieval tasks, grouping together "chunks" of words. The Acronym-meaning extracting machine then looks for a target acronym in a context such as:

EXP_i, EXP_j, T_ACRONYM, EXP_k, EXP_m,

where the expressions are either tagged strings or phrases and T_ACRONYM is another expression (usually a tagged string).

1. [['the', 'DT'], ['performance', 'NN'], ['of', 'IN'], ['an', 'DT'], ['automatic', 'JJ'],['speech', 'NN'], ['recognition', 'NN'], 'NX'],
2. ['(', '('],
3. [['ASR', 'NN'], 'NNX']
4. [')', ')']

The above example shows 4 expressions that are the input for the Acronym-meaning recognizer. Under such a configuration, the strings: 'The performance of an automatic speech recognition' and 'ASR' will be used as input to the regular expression machine for Acronym-meaning pair recognition.

This design allows us to highly constrain the context within which to search for the acronym expansion. In an algorithm that considers only the strings and their context, an arbitrary window or boundary must be set. This arbitrary boundary allows more errors, as can be seen in the comparison with *Acrophile* below. With shallow parsing, the boundary is established naturally by the properties of the language. With this strategy, the meaning or expansion is specified to be a noun phrase that is close to the target acronym. Constraints can be stated on which are the possible other expressions in

the context of a target acronym, i.e., punctuation marks, noun phrase coordination.

We use finite state automata which consume the expressions, checking their types (e.g. Noun phrase, verb phrase, punctuation symbol). If a configuration is found with a target acronym and a target expansion expression, then the strings corresponding to both expressions are supplied to the string acronym finder (the previous strategy), which will decide if a substring of the target expression matches the acronym. If so, it will be scored as a positive identification and stored in the acronym database.

The technical notions of *precision* and *recall* which are a standard in Information Retrieval technologies, clearly apply to this task in the following way (see [4] [6] for similar definitions):

$$\textbf{Precision} = \frac{\text{\# of correctly retrieved acronym-meaning pairs}}{\text{\# total of retrieved acronym-meaning pairs}}$$

$$\textbf{Recall} = \frac{\text{\# of correctly retrieved acronym-meaning pairs}}{\text{\# total of acronym-meaning pairs in the data}}$$

The Tests

The following tests were performed in different steps.

Test #1. The Development Corpus

The first results were obtained using the development corpus and the *regular expression* algorithm. From this corpus, ACROMED retrieved 123 pairs, of which 106 were correct (the Gold Standard had 149 pairs). The measures of precision and recall (specified in Table I) were comparable to the results of others reported in the introduction, which were intended for use on unrestricted text. Here we used basically the same technology as those previous approaches, but adapted it to the particular characteristics of the biomedical domain to take into account the kind of characters we showed in the examples above. The results using the development corpus and the *syntactically constrained* algorithm improved significantly regarding precision. Moreover, recall was not compromised by this.

Test #2 The Evaluation Corpus

The *regular expression* algorithm retrieved 117 pairs from the Evaluation Corpus of which 106 were correct pairs. Using the *syntactically constrained* algorithm the following results were obtained with the evaluation corpus: 105 pairs were retrieved by ACROMED. From those pairs 104 were correct pairs, and 1 did not match exactly the items in the Gold Standard. In this case, **TH** was assigned the meaning "helper T" and it was annotated as "CD4 helper T" (the phrase was: " that is mediated by the activation and differentiation of CD4 helper T (TH) cells into TH1 and TH2 effector cells. "). Thus, this hit was not a false positive, but a partial retrieval of an alias/acronym hybrid. The total number of pairs in the Gold Standard were 168. This corresponds to the precision and recall measures seen in Table 3.

Table 2 Precision and Recall

Development Corpus 149 pairs	**Precision**	**Recall**
regular expression	88.1%	73.2%
syntactic constraints	97.2%	72.5%

Table 3 Precision and Recall

Evaluation Corpus : 168 pairs	**Precision**	**Recall**
regular expression	90%	63%
syntactic constraints	99%	61.9%

Consistent with the results of others, the results with regular expressions suggest that a maximum in the usual trade-off between precision and recall is limited to around the 90% precision using finite state machinery. If precision is improved there is a loss in recall and vice versa. The results with the Evaluation Corpus show reduced recall, which should be expected, given that the *regular expression* algorithm was highly tuned for this Developmental Corpus. Strikingly, however, the precision that was obtained with the *syntactic constraints* algorithm was not diminished when applied to this corpus. This result shows that the syntactic machinery, which was not specifically designed for this task, but instead is part of a suite of tools for retrieving information from the Medline database is responsible for the improved precision here.

Test #3 A Measure of Comparison.

In order to directly compare the performance of ACROMED to *Acrophile*, we evaluated the performance of *Acrophile* on our biomedical Gold Standard texts. According to the results reported by [1] and summarized in the introduction, we determined that *Acrophile* would be a good measure of comparison. It was designed and tested with a considerable corpus of unrestricted text. We submitted both the Development Corpus and the Evaluation Corpus to the *Acrophile* server using both their Contextual/Canonical algorithm that was reported to have better performance and the Contextual algorithm that should

have better capabilities to handle the kind of data in Medline. The results are summarized in the following table.

Table 4 Acrophile Performance

Rt.: # Retrieved, Pr : Precision, Rc : Recall.

	Canonical/ Contextual			Contextual		
	Rt	Pr	Rc	Rt	Pr	Rc
Development Corpus (149)	34	100	23%	99	86%	57%
Evaluation Corpus (171)	49	90%	26%	81	85%	40%

The results from testing our corpora with Acrophile, show that the performance of a general purpose Acronym-meaning retrieval system is greatly diminished when applied to the kind of data that is found in the Medline abstracts. Strikingly; the recall performance is from 88% to 57% and 40%. Also it shows that both precision and recall are lower in the Evaluation Corpus which seems to be "harder" than the Development Corpus we used.

Both *Acrophile* and our *regular expression algorithm* produced some false positives (errors in the acronym meaning pair) which are clearly related to the problem of assigning a window or boundary for the search of the long form of the acronym. Examples of this are:

RNA = repeat number to about, extracted from: "of an essential subunit of RNA polymerase I (Pol I) in rpa135 deletion mutants triggers a gradual decrease in rDNA repeat number to about one-half the normal level."

p16 = products, extracted from: "which encodes two gene products (p16(INK4a) and p19(ARF))".

Conclusions

The results we presented show that a considerable gain in precision is achieved if syntactic information is used to constrain the context in which to search for the long form of an acronym. Although recall is lower in the Evaluation Corpora, this is largely due to the limited pattern-matching machinery we were using (only looking at the lefthand context). The shallow parsing and pre-processing machinery might introduce some errors, or noise, but it seems that its effects on recall are not significant. Our next steps will be: (1) to include broader configurations (i.e., additional patterns not only targetting an acronym within parentheses and its possible meaning to its left) where acronym-meaning pairs are possible; and (2) to make finer grain distinctions in the corpus annotations to distinguish between aliases and acronyms.

Acknowledgments

This work was supported by NIH grant R01-LM06649 to James Pustejovsky at Brandeis University and Brent Cochran at Tufts University.

References

[1] Andrade, MA et al. *Automatic annotation for biological sequences by extraction of keywords from MEDLINE abstracts. Development of a prototype system.* AAAI (American Association for Artificial Intelligence, www.aaai.org), 1997.

[2] Blasche C et al. *Automatic extraction of biological information from scientific text: protein-preotein interactions.* AAAI, 1999.

[3] Craven M and Kumlien J *Constructing Biological Knowledge Bases by Extracting information from Text Sources.* In Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology, 1999.

[4] Larkey, LS et al. *Acrophile: An Automated Acronym Extractor and Server.* To appear in DL00, Association for Computer Machinery Inc.

[5] Rindflesch T et al. *Extracting Molecular Binding Relationships from Biomedical Text.* In Proceedings of the 6th Applied Natural Language Processing Conference. Association for Computational Linguistics, 2000.

[6] Taghva, K et al. *Recognizing Acronyms and their Definitions.* Technical Report 95-03, ISRI (Information Science Research Institue) UNLV, 1995.

[7] Yeates, S *Automatic extraction of acronyms from text.* In Proceedings of the Third New Zealand Computer Science Research Students' Conference. University of Waikato, 1999.

[8] Yeates, S et al Using Compression to identify acronyms in text. Submitted to Data Compression Conference. DCC, 2000

Address for correspondence

Jose M. Castaño jcastano@cs.brandeis.edu