Extraction and Disambiguation of Acronym Meaning-Pairs in Medline

James Pustejovsky[a], José Castaño[a], Brent Cochran[b], Maciej Kotecki[b],
Michael Morrell[a], Anna Rumshisky[a]
[a]*Laboratory for Linguistics and Computation at Brandeis University,*
415 South St., Waltham, MA 02454
PH: +1-781-736-2709 FX: +1-781-736-2741
{jamesp,jcastano,mmorrell,arum}@cs.brandeis.edu
[b] Department of Physiology at Tufts University,Tufts
University School of Medicine, 136 Harrison Ave., Boston, MA
PH: +1-617-636-0442 FX: +1-617-636-6745
{cochran, mkotecki}@opal.tufts.edu

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. Moreover, an even harder problem is sense disambiguation of acronyms; that is, where a single acronym, termed a polynym, has a multiplicity of meanings, a common occurrence in the biomedical literature. In such cases, it is necessary to identify the correct corresponding sense for the polynym, which is often not directly specified in the text. Here we present a system called **Acromed** which finds acronym-meaning pairs as part of a set of information extraction tools designed for processing and extracting data from abstracts in the Medline database. Our strategy for finding acronym-meaning pairs differs from previous automated acronym extraction methods by incorporating shallow parsing of the text into the acronym recognition algorithm. The performance of our system has been tested with a highly diverse set of Medline texts, giving the highest results for precision and recall, thus far in the literature. We then present **Polyfind**, an algorithm for disambiguating polynyms, which uses a vector space model. Our disambiguation tests produced 97.62% accuracy in one test (on acronyms) and 86.6% accuracy in another (on aliases).

1 Introduction

The use of computational techniques to automatically extract information from biomedical abstracts, and in particular from MEDLINE, has received much attention recently (e.g., Andrade *et al.* (1997), Blaschke *et al.* (1999), Craven *et al.* (1999), Rindflesch *et al.* (2000)). Most of this work has focused on named entity extraction and verb-specific relational extraction from texts. Progress with general purpose information extraction has been slower than expected because of the peculiar demands that biomedical texts place on natural language processing systems. Acronyms pose an interesting challenge to information extraction systems for several reasons. First, the problem of retrieving the meaning of acronyms in text is related to that of entity identification, since it is necessary to know which entity an acronym expression refers to in a text in order to accurately identify and extract the interactions and relations between terms as expressed in the abstract. Secondly, the available databases for acronyms and abbreviations for the biomedical domains are incomplete; for example, the current UMLS database contains 10,410 entries (considering polynyms as separate entries);

similarly, acronym dictionaries are out of date by the time they are published (e.g., Jablonski, 1998). These represent only a portion of the acronyms occurring in the Medline corpus. For example, running our system over 40,956 abstracts (one month of MEDLINE database) resulted in 9,272 unique acronyms which were not identified in the UMLS database[1].

1.1 Determining the meaning of acronyms automatically

The problem of automatically determining the meaning of acronyms in biomedical texts is both a critical and 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 novel acronyms and new acronym meanings. Furthermore, 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 (Larkey *et al.* (2000), Taghva *et al.* (1995), Yeates (1999), Yeates *et al.* (2000)). Most of these works distinguish between "standard" acronyms on the one hand, and abbreviations, aliases, and short acronyms on the other. As a result, these strategies miss many important terms in the text that are unidentified and hence ignored. In the present work, the algorithm and results refer to all forms of acronyms, as well as abbreviations, and aliases to a certain extent. We assume that the task an automated system must solve in this case is to identify short-form–long-form pairs, where there exists a mapping from characters in the short form to characters in the long form, of whatever form.

For instance (Larkey *et al.* 2000) and Taghva *et al.* (1995) do not consider *acronyms* such as the following:

CFDA: *carboxifluorescein diacetate*
PMA: *phorbol ester 12-myristate-13-acetate*
TE: *trophectoderm*

Such short forms do not constitute "acronyms" by the definition established by these authors. The best results reported from these techniques are summarized in Table I below.

Table 1. Precision and Recall from free text[2].

System	Precision	Recall
[1] Canonical/ Contextual	87%	88%
[1]Canonical	96%	60%
[1]Canonical Simple	94%	59%
[2]Pattern matching	68.68%	91.91%
[3]Data Compression	90.9%	80.8%
[4] LCS Algorithm	98%	93%

The results from Taghva *et al.* (1995) look impressive (with 93% recall and 98% precision), due in part to the constraints mentioned above; namely, their algorithm does not consider acronyms with fewer than three characters, and also excludes abbreviations such as *DOP* (dioctyphthalate), *TRU* (transuranic) and *MW-hr* (megawatt-hour). These kinds of short forms are very common in the Medline corpus, and is a major factor, for example, for the poor performance of Larkey *et al.*'s system, Acrophile, when run over the Medline corpus, as we show in Section 3.2. Although some of these results are good, they are far from optimal and we have found that their performance is significantly worse when applied to biomedical texts. Since our goal is to automatically populate databases and develop intelligent search and navigation algorithms for biomedical texts, much higher precision is required.

[1]This experimental run generated the recognition of 27,292 acronym-meaning pairs, 12,970 of which resulted in unique pairs and 2,896 ocurred more than once in the database.

[2]Number 1 corresponds to Larkey *et al.* (2000) number 2 to Yeates (1999), number 3 to Yeates *et al.* (2000) and number 4 to Taghva *et al.* (1995)

In this paper, we analyze the performance of new strategies we have designed to improve both the precision and recall of automated acronym identification in biomedical texts. As a first step, we 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. As a second step, we constrained and circumscribed 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 sentence within which a potential acronym may occur, we apply a finite-state matching algorithm with considerable precision to identify the long form. Also, by having identified the syntactic structure containing a target acronym and its immediate environment, we are able to examine much richer contexts, improving our recall. 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:

1. Determining the window size (or context) of the text within which the long form for the acronym lies.
2. Identifying the long form by matching, deleting and simplifying character strings relative to the acronym itself.

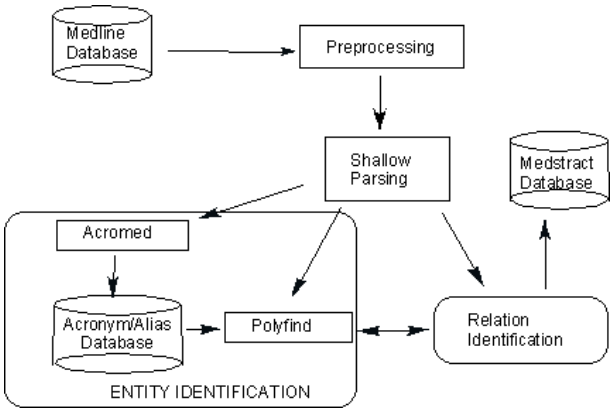
We show that much greater accuracy can be attained if these two problems are treated as separate computational tasks. Importantly, the first problem is solved by a constrained context-free parsing algorithm, developed independently for the automated interpretation and extraction of protein and gene descriptions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

1.2 Acronym Sense Disambiguation

The problem of sense disambiguation is a crucial one in an information retrieval system (cf. Krovetz *et al.* (1992), Sanderson (1994), Krovetz (1997)). For instance, if the task is to retrieve documents related to SRF, as in the sense of "Serum Response Factor", then those documents containing the string SRF and a different meaning from Serum Response Factor (such as "Spatial Receptive Field"), should not be retrieved. Often it is not possible to perform this task using query reformulation with Boolean expressions given there is no long form in the document to disambiguate the meaning of the acronym. The problem that acronyms with multiple senses (referred to here as *polynyms*) present has not been previously addressed in the literature dealing with automated acronym-meaning discovery. We have implemented and tested the performance of a vector space model (cf. (Salton 1971)) designed to disambiguate the meaning of biomedical polynyms. The results, which demonstrate the utility of this method, are surprisingly accurate and are discussed in Section 4.2.

1.3 General Architechture

Both the acronym identification algorithm (Acromed) and the polynym disambiguation algorithm (Polyfind) are embedded in a larger biomedical information extraction system called Medstract. An illustration of the architecture is shown below.



Preprocessed text is fed into Acromed where acronyms are associated with corresponding long forms and used to populate an acronym database. Acronyms without any associated long forms are processed by Polyfind, to associate the correct meaning for the short form. The results of these processes are used by the relation identification algorithms to more accurately populate the Medstract database.

2 ACROMED: Materials and Methods

2.1 General Design of the Experiments

We assume that acronym-meaning identification is a subclass of the alias identification problem, and that acronyms are a subtype of aliases. The goal of the task presented in this paper is to capture the meaning of those acronyms which are introduced in the text. There are two possibilities: either the acronym has a long form in the text, or it doesn't. In the first case, if the acronym is introduced with its long form Acromed might find it. If the long form is not found, the meaning of the acronym will be looked for in the acronym database. If any of the meanings corresponding to the acronym are found in the immediate context, then it will be assigned to the acronym. Otherwise, Polyfind will be used to disambiguate the possible meaning. In general, this is the case with acronyms assumed to be known by the reader and are not the target of the task to be performed by Acromed: e.g., if the acronym "HIV" occurs in an abstract without its long form ("Human Immunodecifiency Virus") in its proximal context, it is a target for Polyfind and not for Acromed. It is assumed in this case that the knowledge of what the entity "HIV" stands for must be found independently of the occurrence of a long form.

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 the occurrences of aliases pairs by a biomedical specialist. It contains 155 occurrences of Acronym-meaning pairs. This was established as the Gold Standard for our development corpus.[3] We used the development corpus to test and improve our set of regular expression strategies.

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

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".

The last 3 of these examples are clearly aliases that are not acronyms, whereas ASR is a straightforward acronym. An accounting of the alias pairs reveals that only 6 of the 165 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` relation.

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,[4] 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.

3 The implementation

Our strategy for extracting acronym-meanings (also called "long forms" or "expansions") in the Medline database was developed following two different strategies. First we considered the problem of recognizing acronym-meaning pairs as the problem of finding two strings in a text that satisfy certain properties, such as matching certain regular expressions. We call this the *regular expression algorithm*. The input text is a simple sequence of strings. This is basically the same strategy used by the authors mentioned in the previous section. We designed regular expressions matching potential acronyms 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, then it is rated with a formula that compares how good the acronym is compared to a 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 the characters contained within it are tested as a match for 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 number of words in the match[5] (see Taghva *et al.* (1995) for a similar approach). 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

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

[5] The corresponding formula is:

$$\text{Score} = \frac{\text{\# of words in the match (not including stopwords)}}{\text{\# of characters in the acronym}}$$

used here, but which would be expected to improve the recall, would be to add the capability of looking to the right 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 determine 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 completely 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, it groups together "chunks" of words. The Acronym-meaning extraction 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) as in:

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 four expressions that are the input for the Acronym-meaning recognizer. Under such a configuration, the two 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 a finite-state automaton which consumes 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 first experiments with the *syntactic constraints* machinery used only the following pattern (where T_LF means target where to find the long form and T_A means target or possible acronym):

1. T_LF_Noun Phrase_1 (T_A_ Noun Phrase_2)

The following experiments using the *modified syntactic constraints*, had these additional patterns:

1.' T_A_Noun Phrase_1 (T_LF_Noun Phrase_2)
2. T_A_Noun Phrase_1, T_LF_Noun Phrase_2
3. (T_A_ Noun Phrase_1) T_LF_Noun Phrase_2

We also developed another regular expression pattern, which by itself resulted in lower precision; but these regular expression patterns were used only in the syntactic environment (1) above. This allowed us to increase considerably the recall measure with only a small decrease in precision. The technical notions of *precision* and *recall* which are a standard in Information Retrieval technologies, clearly apply to this task (see

Larkey *et al.* (2000), Yeates (1999) for a similar view).[6]

3.1 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, 106 of which 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 these previous approaches, but adapted it to the particular characteristics of the biomedical domain to take into account the kinds od idiosyncracies we showed in the examples above. The results using the development corpus and the *syntactically constrained* algorithm improved significantly regarding precision. Moreover, it is interesting that recall was not compromised by this move.

Test #2 The Evaluation Corpus

The *regular expression* algorithm retrieved 117 pairs from the Evaluation Corpus, 106 of which 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 following precision and recall measures:

Table 2 Precision and Recall

	Development Corpus		**Evaluation Corpus**	
	Precision	Recall	Precision	Recall
regular expression	88.1%	73.2%	90%	63%
syntactic constraints	97.2%	72.5%	99%	61.9%
modified syntactic constraints	94.6%	82.5%	98.3%	72%

Consistent with the results of previous work, our 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.

[6]The corresponding definitions are:
Precision = number of CR-AMP / number of TR-AMP;
Recall = number of CR-AMP / number of AMP in the data.
where CR-AMP = correctly retrieved Acronym-Meaning pairs; TR-AMP = Total retrieved Acronym-Meaning pairs; and AMP = Acronym-Meaning pairs.

3.2 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 [7]. 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 3 Acrophile Performance

	Canonical/Contextual		**Contextual**	
	Precision	Recall	Precision	Recall
Develo pment Corpus (149)	100%	23%	86%	57%
Evaluati on Corpus (171)	90%	26%	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. Furthermore, 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." and

p16 = products, extracted from: "which encodes two gene products (p16(INK4a) and p19(ARF))"
These problems can be overcome when syntactic contexutal information is provided through the parser.

4 Polynym Disambiguation

4.1 Polynym Normalization

Once an acronym is found it is looked up in the acronym database, if the acronym-meaning pair is already stored there; if so, it stores the corresponding data in the database to index the occurrence of the acronym. If not, it is compared with the existent polynyms for the same acronym (if any). If the "meaning" or long form is equivalent to the meaning of any existent polynym, an equivalence class of acronyms is created. The equivalence classes are determined by morphological variances in the long form, capitalization versus lower case, dashes versus space, plural versus singular, or substring properties (one of the long forms is a prefix or a sufix of the other. Once equivalence classes are created, a new acronym is compared to all equivalence classes. If a class is found where one element is equivalent to the new acronym, then the acronym is added to that class. Otherwise, a new class (with one member) is created.

[7] We also peformed another test with Acrocat (http://www.aclweb.org/ bwhitman/acrocat/). There is no documentation on how Acrocat works from the files accessible from the web page, except for a very short description. The results were significantly worse than those of Acrophile, with an estimate of 50-60% precision and 20% recall.

4.2 POLYFIND

Recall that a *polynym* is an acronym or alias that has several possible long forms or meanings associated with it. For instance, Larkey *et al.* (2000) mentions 11 expansions for the acronym **EWI**[8], however they do not address the actual disambiguation problem of *polynyms*. That problem is as follows: if a *polynym* is found in a text, and the meaning is not available or defined in that text, we must have some way to determine the current long form (meaning) for the free-standing *polynym*. We conducted two preliminary polynym disambiguation experiments, one for disambiguating acronyms, and another for disambiguating aliases. In both cases, we tested the performance of the standard variations of the vector space model Salton (1971), using the SMART Information Retrieval System, Version 11.0 Smart (1999).

4.2.1 Disambiguating an acronym with multiple senses

We selected the acronym **SRF** with 10 distinct meanings, and collected all Medline abstracts defining **SRF** for each of its meanings (i.e., those abstracts containing the acronym and the long form).[9] The abstracts defining each meaning were pulled together, the resulting 10 abstract sets to be used as document templates against which the ambiguous occurrences of SRF were to be evaluated. We further collected 42 abstracts in which this acronym was used without the long form. These occurrences of the **SRF** acronym were then manually disambiguated, and split into four groups, according to the intended meaning: *Serum Response Factor*, *Subretinal Fluid*, *Surfactin*, and *C Elegans surface antigen gene mutations*.

The abstracts from each of these 4 groups were then used as queries, and evaluated against the 10 meaning sets. In order to remove the confounding effect of the long form portions that may have occurred elsewhere in the abstract used as a query, all the words comprising the long forms of SRF were removed from the vectors. Both the query and the document template vectors included the values for the tokens from the title, the authors' names, the journal, and the body of the abstract.

We used a standard within SMART 'atc' variation of the basic 'tf*idf' weighting scheme, using *augmented norm* for *term weighting*, and *cosine normalization* for the *vector length* (Smart (1999), Salton *et al.* (1988)). The measure of similarity between the query and each of the meaning sets was then computed, and the query was considered to have been disambiguated correctly if the highest similarity score was obtained by the set with the correct SRF meaning. Under this scheme, we obtained 97.62% accuracy in disambiguation.

4.2.2 Disambiguating aliases with Multiple Senses

For this experiment, we selected the alias **p21** with three commonly occurring meanings: (1) **p21 Waf1**, also called **Cdi** and **Sdi**, (2) **p21 Ras**, and (3) **p21** designating a chromosomal band on the short arm of human chromosome 21.

We collected 1,500 abstracts containing the string "p21" where the corresponding meaning of p21 was identified manually. Each abstract was assigned to the corresponding meaning set. The resulting 3 sets each contained approximately 500 abstracts. We used the procedure similar to the one described above for SRF. Thirty abstracts were removed from each set to serve as the total of 90 queries. Again, the alias was considered to have been disambiguated successfully if the highest similarity score was obtained by the appropriate comparison set.

It is important to note that the first two meanings of **p21**, namely **p21 (WAF)** and **p21 (RAS)**, come from the same field and therefore are closely related. Consequently, their disambiguation should be expected to be more difficult using the methodology from above. The following steps were then taken in an attempt to

[8]The forms were: Edison Welding Institute, Education With Industry, Electronic Warfare Intelligence, Equal Width Increment, Explosive Waste Incinerator, Eijkman Winkler Institute, European Web Index, European Wireless Institute, Edison Welding, Electro World Inc, Executive Women International.

[9]The meanings, and the number of abstracts found in each case, are: Serum Response Factor, 30; Subretinal Fluid, 31; Subretrofacial Nucleus, 17; Mutants of C Elegans, 6; SRF-3 mutant SFV virus, 4; Spatial Response Function, 3; Strained Rumen Fluid, 2; Surfactin, 2; Secretin Releasing Factor, 1; h-SRF Human Somatotropin Releasing Factor, 1.

increase precision. The first 50 most frequent tokens, including the actual meaning strings, were removed from the vectors. We used three different scoring methods, atc, ntc, and atn: the last two were selected in order to remove the additional vector-length normalization that has been claimed to lead to poorer results in cases of short documents (cf. for example, Salton *et al.* (1988)). Additionally, we modified the number of abstracts used for the comparison set. We then conducted the disambiguation test using (a) 100-abstract comparison sets, and (b) 10-abstract comparison sets, the latter allowing us to simulate the conditions for less frequently occurring *polynyms*.

With the larger collection of data (100 abstracts) used for the comparison sets, the ntc scoring method gave best results giving 82.22% accuracy. For the smaller collections of data (10 abstracts per comparison set), the atn scoring method gave higher accuracy: 70%.

These figures seem quite modest, compared to the results for the acronym, but that is to be expected, considering how close two of the senses of p21 are. For instance, if one of the close sets is removed from consideration, the accuracy figures rise considerably: smaller comparison set (10 abstracts): 83.3% accuracy (atn) and larger comparison set (100 abstracts): 86.6% accuracy (atn).

Conclusions and Discussion

Regarding Acromed, the results we have presented show that a significant gain in precision is achieved if syntactic information is used to constrain the context within which to search for the long form of an acronym. Initially, recall was lower in the Evaluation Corpus, due to the limited pattern-matching machinery we were using (only looking at the lefthand context); subsequently, however, when we included right context as well, the recall increased considerably. The shallow parsing and pre-processing machinery may introduce some errors, or noise, but it seems that its overall effects on recall are not significant. Our next steps will be to improve the regular expression machinery to get higher recall if possible. The improvement demonstrated by this technique over the existing algorithms for acronym identification is both considerable and encouraging, given the importance of this task for the extraction of information from biomedical texts.

Regarding Polyfind, the preliminary results presented here demonstrate that the application of a vector space model for polynym sense disambiguation is both workable and promising. These results, however, do not take into consideration any syntactic information, which may eventually be required for identifying the correct meanings for alias forms with high accuracy. This is part of our current focus of research in this area.

Acknowledgments

This work was supported by NIH grant R01-LM06649 to both James Pustejovsky and Brent Cochran.

References

Andrade, Miguel A. and Valencia, Alfonso. Automatic annotation for biological sequences by extraction of keywords from MEDLINE abstracts. Development of a prototype system. AAAI, 1997.

Blasche, Christian; Andrade, Miguel A.; Ouzounis, Christos and Valencia, Alfonso. Automatic extraction of biological information from scientific text: protein-preotein interactions. AAAI,1999.

Buckley, C. Implemetation of the SMART Information Retrieval System. Technical Report 85-686, Cornell University, Computer Science. 1985.

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

Jablonski, Stanley (Editor). Dictionary of Medical Acronyms and Abbreviations, 1998.

Krovetz, R. and Croft, W. Lexical ambiguity and information retrieval. ACM Transactions on Information Systems, 10(2):115-141, 1992

Krovetz, R. Homonomy and polysemy in information retrieval. In 35th Meeting of the Association for Computational Linguistics and the 8th Meeting of the European Chapter of the Association for Computational Linguistics (ACL/EACL-97), pages 72-78, Madrid, Spain, 1997.

Larkey, Leah S.; Ogilvie, Paul; Price, M. Andrew and Tamilio, Brenden. Acrophile: An Automated Acronym Extractor and Server. To appear in DL00, Association for Computing Machinery Inc.

Ohta, Yoshihiro; Yamamoto, Yasunori; Okazaki, Tomoko; Uchiyama, Ikuo and Takagi, Toshihisa. Automatic Construction of Knowledge Base from Biological Papers. AAAI, 1997.

Rindflesch, Thomas C; Rajan, Jayant V. and Hunter, Lawrence. Extracting Molecular Binding Relationships from Biomedical Text. In *Proceedings of the ANLP-NAACL 2000*, pages 188–195 Association for Computational Linguistics, 2000.

Salton, G. The SMART Retrieval System: Experiments in Automatic Document Processing. *Prentice Hall, Englewood Cliffs*, NJ. (1971).

Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24(5):513-523, (1988).

Sanderson, M. Word sense disambiguation and information retrieval. In Proceedings, ACM Special Interest Group on Information Retrieval, pages 142-151,1994.

The SMART Information Retrieval System, Version 11.0. *Cornell University Computer Science.* ftp://ftp.cs.cornell.edu/pub/smart/. (1999).

Strzalkowski, T. Information retrieval using robust language processing. In AAAI Spring Symposium on Representation and Aquisition of Lexical Information, pages 104-111, Stanford, 1995.

Taghva, Kazen and Gilbreth, Jeff. Recognizing Acronyms and their Definitions. Technical Report 95-03, ISRI (Information Science Research Institue) University of Nevada, Las Vegas. 1995.

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

Yeates, Stuart; Bainbridge, David and Witten, Ian H.. Using Compression to identify acronyms in text. Technical Report 00/01 Cs Dept. University of Waikato, New Zealand. Submitted to Data Compression Conference. DCC, 2000.