A Biological Named Entity Recognizer

M. Narayanaswamy, K. E. Ravikumar, K. Vijay-Shanker

Pacific Symposium on Biocomputing 8: 427-438(2003)

A BIOLOGICAL NAMED ENTITY RECOGNIZER

MEENAKSHI NARAYANASWAMY AND K. E. RAVIKUMAR
AU-KBC Research Centre
Chennai 600044 INDIA

K. VIJAY-SHANKER
Department of Computer and Information Sciences,
University of Delaware
Newark, DE 19716, USA

In this paper we describe a new named entity extraction system. Our system is based on a manually developed set of rules that rely heavily upon some crucial lexical information, linguistic constraints of English, and contextual information. This system achieves state of art results in the protein name detection task, which is what many of the current name extraction systems do. We discuss the need for detection of chemical names and show that we not only obtain a high degree of success in recognizing chemicals but that this task can help improve the precision of protein name detection as well. We use context and surrounding words for categorization of named entities and find the results obtained are encouraging.

1 Introduction

Research in biology in the past decade has generated a large volume of biological data that is available only in the literature, i.e. in the textual form, which are stored in databases such as MEDLINE. The quality and amount of continuously updated knowledge makes this an extremely useful form of information that needs to be trapped for research and development. There is a need to develop automated intelligent text analysis tools in order to extract useful information from the literature.

As a first step towards information extraction on various interactions between biological entities, a system must identify biological enities, e.g. gene, protein, chemical, cell and organism names. Handling of unknown words, long compounding of word sequences are some of the difficulties in name recognition in this domain. A few biological name entity recognition tools (e.g., Collier et al.[2], Friedman et al.[3] Fukuda et al.[4]) exist which were developed as an integral part of different information extraction applications. Much of these efforts are geared to extract protein and gene names. Some of the tools use statistical or machine learning approaches in identifying names from biological texts.

1.1 Main Features of Our System

Our approach is a symbolic one and based on a set of manually developed rules. These rules exploit surface clues and simple linguistic and domain knowledge in identifying the relevant terms in the biomedical papers. While our approach is not based on machine learning or statistical methods, we wish to point out that the results we obtain are due to the type of features we extract from text and that these can just as easily be exploited in a statistical or machine learning setting.

There are advantages and disadvantages of using either a manually designed system or one based on statistical or machine learning. A statistical or (supervised) machine learning approach requires large amount of text where the named entities are marked. One of the chief reasons for not having used a supervised machine learning approach is because of a lack of an annotated corpus. Creating an error-free annotated corpus will take extensive time and effort. Besides, we have noticed that there doesn't appear to be an exact consensus on exactly what constitutes a name (even when we limit it to proteins/genes). Additionally, as more information extraction work proceeds, the need to extract named entities of other types might arise (for instance, we discuss below why it might be useful to extract chemical names as well). So until these matters are better settled, we decided not to take a supervised learning approach (to avoid reannotations and retraining of models).

In many respects, our approach is inspired by the design of PROPER (PROtein Proper-noun Extracting Rules), also based on a manually designed set of rules (Fukuda et al.[4]). We believe our work extends the term recognition capability of this system, and other systems, in two significant ways. First, we believe that we improve upon the precision and recall (standard measures for evaluating name/term recognition systems). Second, while most systems tend to focus on recognizing protein/gene names alone, we recognize other types of names/terms as well. In particular, we also recognize chemical names. While the work in the GENIA project (see e.g., Collier et al.[2]) is also meant to recognize terms beyond names and proteins, they do not identify chemical names. There are two major reasons why we felt it was important to capture chemical names. First, biomedical papers often contain plenty of chemical names. These chemical names do share various features that are used to identify protein names. Hence, we believe name detection systems can wrongly identify chemical terms as protein names. Our approach then is to identify the chemical names and classify them as such and thereby improve the precision of the (protein) name recognition. Second, most of the current work on information extraction from biomedical papers is geared towards extracting protein-protein interactions. Hence most of the biological named entity recognizers are geared towards identifying protein and gene names only. In biological interactions, the entities involved are not just proteins and genes but chemicals too. From information perspective, which might be important in aiding drug discovery process it is very crucial to identify chemicals in the text. We are only aware of work by

Wilbur et.al.[5] for extracting chemical names automatically. Existing tools mostly use a dictionory or ontology lookup to identify chemical and drug names.

In our name recognition module we intend to capture the following biological terms and classifying them under their respective category:

* **Protein/gene** – Terms that correspond to names of proteins or genes. Like most other systems, but unlike Hatzivassiloglou et al.[6], we do not distinguish between protein and gene names.

* **Protein/gene parts** - Terms corresponding to protein/gene parts. Examples of these include MH2 domain, and lysine residue. Protein name detection tools such as KeX (Fukuda[4]) which is based on PROPER do not attempt to distinguish protein names from terms of this category.

* **Chemical** – Examples include Indomethacin, N-methylformamide, and suberoylanilide hydroxamic acid**.**

* **Chemical parts** – Terms, like methyl groups, that correspond to parts of chemicals.

* **Source** - Terms that represent source terms including cells, cell parts and organisms.

* **General** (biological) – Terms that can truly be classified as belonging to more than one class or related to the above class but can not be classified as belonging to one of them. A protein-chemical complex, such as hematoporphyrin-LDL complex, would belong to the latter case.

2. Design of the System

The text (Medline abstracts, in our current experiment) are split into sentences, tokenized and then part of speech tagged using Brill's tagger (Brill[1]). In this section and in Section 4 we describe the main modules of our system. These modules are applied in the order they are described below.

2.1 Identifying abbreviations

Abbreviations are widely used in medical literature. Hence identifying them and their expanded form is a necessary task in biological name entity recognition. Furthermore in our setting we have to associate with each occurrence of an abbreviation its classification (protein/gene, protein/gene parts, chemical, chemical parts and source).

Our system works on the premise that typically the first occurrence of an abbreviation occurs in a pattern in which the original term is followed by the abbreviation within the parenthesis. E.g., *testosterone repressed prostrate message 2 (TRPM-2)*

We have currently implemented a simple algorithm to identify such pairs. It only covers cases where the acronym appears in parenthesis and is to the right of its definition. It works by matching as many characters of the acronym as possible. It currently identifies 94.2% of such pairs in our test set but however has an accuracy of 87.4% only. We are working on improving this component and incorporating ideas from Taghva et. al[7].

Once a pair, say for example *CBP/p300 associated factor (CAF)*, is identified we attempt to classify it (as described later) based on the expanded form *CBP/p300 associated factor*. This type information is associated with the abbreviation so that remaining occurences (within the same abstract/paper) of the abbreviation can be classified appropriately. Note that while future occurrences of the abbreviation *(CAF)* can be detected, it would be difficult not be possible to figure out its classification just based on its surface form.

2.2 Core Terms and Functional Terms

Like in KeX that incorporates PROPER, the basic idea involves identifying two types of terms: core terms (c-terms) and functional terms (f-terms). This categorization typically applies to individual words. C-terms have surface features (such as capital letters, numerals, and special symbols) that are used by most name recognition systems. On the other hand, the set of functional terms are specific to the name recognition in biomedical domain and play a key role in the classification of the extracted terms into our five categories as described below. Perhaps since the latter issue is not considered in KeX, despite making a distinction between c-terms and f-terms, the algorithm does not appear to make distinction between the two types of terms.

2.2.1 C-term Recognition

Like noted above, various surface features of words such as the use of capital letters are used as important clues in most name recognition systems. Following Fukuda et al.[4], we designate words with such surface features as c-terms. However, while these features are useful in identifying names, they clearly do not provide any clue for the appropriate categorization of the identified names. Hence we call the words that contain these features as *general* c-terms (where recall our use of the word general refers to the situation where appropriate categorization is not possible). Names that include general c-terms might however get the appropriate category when they combine with other words (such as f-terms) that provide the necessary information. In addition to the general c-terms we also have two other types of c-terms: *protein* c-terms and *chemical* c-terms. These c-terms contain some information that can be associated with protein/gene names and chemical names respectively.

The recognition of chemical and protein c-terms is obviously not based on clues for names in general. For instance, the module for extracting chemical c-terms includes recognition of chemical root forms. We base this on IUPAC conventions followed in naming chemicals. We also exploit various morphological features and suffixes used extensively in naming chemicals.

For example, consider "We have developed a class of **HDAC** inhibitors, such as ***suberoylanilide hydroxamic acid (SAHA)***, that were initially identified based on their ability to induce differentiation of cultured murine erythroleukemia cells." S*uberoylanilide hydroxamic acid (SAHA)* is a chemical, which inhibits HDAC an enzyme. The suffix –ic followed by acid helps in identifying the two words as chemical c-terms.

In contrast, in the sentence, "Polar organic solvents such as *methanol* or *N-methylformamide* inactivate lipases.", methanol and N-methylformamide are both chemical names. These are first identified as chemical c-terms because they contain the chemical root forms methyl and meth.

KeX identified *N-methylformamide* as protein (it is meant only to recognize protein names) as it has a capital letter. Likewise occurences of SAHA lowers its precision as it misidentified *SAHA* as a protein name. In contrast due to the chemical c-term detection together with the treatment of abbreviations helps us overcome these problems.

We extract protein c-terms purely on the basis of suffixes such as *ase*. Table 1 shows some of the main features for recognizing the remaining c-terms. These features clearly apply to the names of any of our categories.

Table 1 – Word features to capture the c-terms

S.No	Word Feature	Example
1	Caps only	CBP
2	More than one Cap	hCG
3	letters and Digits	TRPM2, H4, p53
4	Single Cap	Aspirin, Asp1
5	Terms having special symbols (/, -, etc)	IL-7 etc

Among the core terms that were selected, some of them are eliminated if they contain only numerals and special characters. Also, special names for experimental techniques and units are eliminated.

2.2.2 Extraction of Functional Terms

Functional words are not only helpful for locating biological terms, they are also very useful for purposes of categorizing the terms properly.

Table 2- Description of functional terms

CLASS	FUNCTIONAL WORDS	DESCRIPTION	EXAMPLE
Protein/Gene	Receptor, protein, factor, gene etc	Protein/Gene/RNA	CREB binding protein
Protein/Gene parts	motif, domain, promoter, etc	Parts of Gene/Protein/RNA	MH2 domain
Chemicals	Steroid, drugs, etc.	Lipid, steroid, organic, inorganic compounds, and carbohydrates	Tertbutyldimethyl silyoxyandrost-4-ene steroid (9)
Chemical parts	Radical, ions, groups etc.	Chemical radicals, inorganic and organic ions etc	acetyl groups, methyl groups etc
Source terms	cells, cell lines, phage	Cell, Organisms, cell parts etc	MCF-10F cells
General terms	Mutants, molecules	Can belong to any of the above category,	asf1mutants Corby mutant

The functional words were similar to the feature terms used by Fukuda et al.[4] However our name recognition tool differs from KeX in dealing with the functional words. The Kex module does not distinguish between the feature terms and the core terms. However we not only distinguish the functional words from the core terms but also classify them under the categories mentioned in the Table 2.

2.3 Concatenation and Extension Rules

So far, we have only identified individual words. Names that might need to be extracted can be several words long. We now consider how such names are extracted. These are done by applying a few concatenation and extension rules.

The first rule we consider says that two f-terms that are next to each other can be combined into one. The category assigned to the new term is determined as follows. If one of the f-terms is of general category and the other is not of general category then the latter's category is adopted. For example, we identify *protein complex* as being of category protein while the f-term complex is of general category.

On the other hand, when both f-terms being concatenated are not general then the category of the one on the right is adopted. This rule is based on the characteristics of the noun phrases in English whose parse structure is typically right branching and whose head is typically rightmost. An example of the application of this rule characterizes *protein fragment* as protein part since the two individual f-terms are of protein and protein part category respectively.

Concatenation of a c-term and f-term is similar. With few exceptions, the f-term is to the right of the c-term and usually determines the type. In *SIR3 Protein fragment*, as argued above, *protein fragment* is a f-term phrase of type protein part. SIR3 is of course a general c-term. Together they form a protein part. There are a few cases where the f-term is of general category in which case the c-term's category provides the necessary information. One such example is *acetyltransferase family*. *Acetyltransferase* is a protein c-term and provides the category information for the composite name.

Concatenation of two c-terms is similar. *H4 acetyltransferase*, and a*cetyltransferase Sas2* are two examples of concatenation of c-terms. The right c-term provides the category information in the first case. In the second example, the right c-term has a general category whereas the left category is more specific and gives the required information.

We now present two examples where a pair of terms being combined are not of general category. The first involves a chemical c-term which combined with a term to its right is no longer of chemical category. (On the other hand, we have rarely noticed this with protein categories.) *Xanthine* is a chemical core because of the chemical root *Xanth*. However, the concatenated phrase *Xanthine oxidase* is designated a protein category c-term phrase because of the nature of the c-term to its right. In *Ras guanine nucleotide exchange factor*, every combination from left to right of two terms leads to a different category until the final assignment of protein category is done.

To conclude this section, we present a few other rules to extend terms beyond individual c-terms. We connect two non-adjacent terms as long as every word in between them is a noun, an adjective or a numeral. This allows us to extract *CREB binding protein* and *MOZ leukemia gene* where the leftmost and rightmost words are already marked (as c-term and f-term) but the one in the middle is not. Also extensions to the left upto a determiner is allowed subject to some conditions and extensions to include greek letters are considered. Finally we have a few rules to drop annotated phrases. For example, a f-term that is not extended to left or right is dropped. Thereby the single word protein by itself is not considered a name.

3. Preliminary Evaluations

To evaluate our system, we collected 55 Medline abstracts, hand annotated them and associated the categories with each marked name. These abstracts were obtained by searching for *acetylates, acetylated* and *acetylation*. This choice was made so that we could have a good proportion of protein, protein part as well as chemical names.

Of the 620 names according to the hand annotation, 593 terms were correctly identified. Thus, for the pure name *detection* task (i.e., without considering the assignment of categories), we get precision, recall and F-measure of 90.39%, 95.64% and 92.94% respectively. These numbers however mask a certain problem that become clear on examination of the following table.

Table 3 – Disambiguation of biological names (preliminary version)

S. No	Total	Terms Disambiguated	Precision	Recall	F-mes
Protein/ Gene	302	104	93.69%	34.44%	50.37%
Protein/ Gene parts	99	43	95.56%	43.33%	59.62%
Chemical	158	116	92.06%	73.42%	81.69%
Chemical parts	13	8	100.00%	61.54%	76.19%
Source terms	36	29	96.66%	80.56%	87.88%

Protein + Protein Parts	401	147	94.23%	40.05%	56.21%

While the precision is still high, the recall falls down significantly. The reason is that for a vast number of cases none of the above categories could be assigned. Of all the terms detected, we can see from Table 3 that only 200 (sum of the second column entries except for the last row – the last row is just the sum of the first two rows) were assigned categories. That is, the category information could not be discerned after the application of the concatenation and extension rules and the term remains categorized as general. In order to compare the performance with other protein name detection system, we should consider the protein and protein parts together (see last row). Since such protein name detectors would consider all the names they detected as protein names, we can also treat all the names detected which our system has not classified as chemical, chemical part or source as protein name. That is, all names that were left in the general category are also considered as proteins (and protein parts). Such an evaluation leads to a F-measure of 86.54 (precision=78.61 and recall=96.26). Notice that the recall has risen sharply. This is because of the 336 terms that were in the general category, 187 protein terms and 52 protein part terms. On the other hand the number of chemical terms was 38. By this strategy, these 38 (although of the 158 chemical terms in the test set) are mistagged as proteins. This causes the precision to fall. Another reason for the reduction in precision is that almost all false positives fell into the general category. And in the situation we were able to identify the category information, the number of false positives is small.

4. PostProcessing Rules

While our system has high precision in disambiguating the terms into the respective classes its recall is low. For many identified names the correct category could not be inferred from the surface string (e.g., SP-A, TAF (II) 30, CD40). We apply additional rules to categorize those names that are in this category with the hope of increasing the recall. These postprocessing rules are essentially of two types. The first considers the surface strings and applies heuristic rules which we were not ready to apply at an earlier stage but are probably more reliable now that we know there is no other clue to categorize it otherwise. The other type that we believe to be novel in the name detection setting is the use of adjoining context for purposes of disambiguation. Hence, given space constraints we will only discuss these.

It is well-known in computational linguistics that words surrounding a phrase (in our case, a detected name) often help in disambiguation. This idea also underlies the work reported by Hatzivassiloglou et al.[6] and Liu et. al[8] who use it for disambiguating ambiguous expressions. Our situation is different in that there is only possible category (from our list) but is unknown at this stage of processing.

We have come up with a list of help-words (h-terms) that are like f-terms in that they provide clue about the category but unlike f-terms are not considered part of the name. A few examples of h-terms are expression, homolog, and recombinant, which are all associated with protein. Although there is much scope of extending this idea, currently we have a small list of h-terms for proteins only. If we observe a *name* and *h-term* adjacent to each other or the pattern *h-term of name* and the name has been classified as general (ior unknown) then we assign the category associated with the h-term to the detected name. Thus, SP-A, TAF (II) 30, and CD40 each get the category protein/gene because they appeared in the context *SP-A expression*, *homolog of TAF (II) 30*, and *expression of CD40* respectively. Despite a preliminary version of such disambiguation rules, as noted below we get very encouraging results. We plan to investigate the use of context in the disambiguation process more thoroughly in the future. We now show our results in a format similar to Table 3. However these results now reflect the application of our postprocessing rules.

Table 4 - Disambiguation of biological names (after post processing)

S. No	Total	Terms Disambiguated	Precision	Recall	F-mes
Protein/ Gene	302	189	96.43%	62.58%	75.86%
Protein/ Gene parts	99	88	97.78%	88.89%	92.75%
Chemical	158	136	93.15%	86.08%	90.86%
Chemical parts	13	11	100.00%	84.62%	91.67%
Source terms	36	35	97.22%	97.22%	97.22%

Only protein terms	401	277	96.85%	69.08%	80.64%

The number of terms which are now in the general or unknown category drops sharply. 102 terms are protein terms after the application of the postprocessing rules as compared to 187 before. However, the biggest change occurs in the protein parts category where we now only have 7 as compared to 52 before. Those in the chemical category drop to 18 from 38 before.

As discussed earlier, we finally categorize all the remaining terms in the general (or unknown) category as proteins or protein parts. Thus in comparison to the results (86.54 for F-measure, 78.61% for precision and 96.26% for recall) obtained before the postprocessing results were applied, we obtain as the final performance results on the Protein Name recognition task as 91.90% for F-measure with 87.92% for precision and 96.26% for recall.

5 Comparison with KeX

The only (protein) name detection system we are able to compare with is KeX which incorporates PROPER, perhaps considered a standard and among the better-known of the existing systems. We could do this because this software is freely available on the internet. Since we are not able to run the other name detection systems on our test set (and neither do we know what their test set was) we are unable to compare our results with them. As the following discussion suggests the nature of test set can make a substantial difference in the results and we didn't choose to compare with the results cited in papers as well.

Our system substantially outperforms KeX on the protein name recognition task when we applied it to our test set. A fairly large proportion of the difference in precision can be attributed to the presence of chemical and source names. The results reported in Fukuda et al.[4] were obtained on a set of Medline abstracts based on SH2 and SH3 domains and perhaps these abstracts didn't have the same proportion of chemical names. And perhaps our test corpus, which was based on search for keywords related to acetylation, might have a higher proportion of chemical names than a randomly chosen Medline abstract.

These results do show two critical points: even within Medline abstracts, results of KeX has varied considerably and also that considering chemical name detection can improve precision of protein name detection. Of course, one of our main motivations for chemical name detection is to aid extraction of information involving chemical compounds and drugs.

We are currently working on improving our acronym-definition detection component; generalize and systematize our approach embodied in the postprocessing rules for assigning categories to the name detected when no surface clue in the name exists; and increasing the categories of named entities we recognize.

References

E. Brill, “Some Advances in Transformation-Based Part of Speech Tagging” in *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-‘94),* Washington 1994

N. Collier, C. Nobata, and J. Tsujii, “Extracting the names of genes and gene products with a hidden Markov model” in *Proceedings of the 18th International Conference on Computational linguistic*s (*COLING 2000)*, Saarbr¨ ucken, 2000.

C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky, “GENIES: a natural language processing system for the extraction of molecular pathways from journal articles” *Bioinformatics*. **17 Suppl 1**(2001)

K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi, “Information extraction: Identifying protein names from biological papers” in *Proceedings of the Pacific Symposium on Biocomputing '98 (PSB'98)*, Hawaii, January 1998.

W.J. Wilbur, G.F. Hazard, G. Divita, J.G. Mork, A.R.Aronson, and A.C. Browne, “Analysis of biomedical text for chemical names: a comparison of three methods” in *Proc. AMIA Symp 1999*, Washington, 1999.

V. Hatzivassiloglou, P.A. Duboue, and A. Rzhetsky, “Disambiguating proteins, genes, and RNA in text: a machine learning approach” *Bioinformatics*.**17 Suppl 1**(2001)

K. Taghva and J. Gilbreth, “Recognizing acronyms and their definitions”, IJDAR. **1**. 1999.

H. Liu, Y.A. Lussier, and C. Friedman, “Disambiguating ambiguous biomedical terms in biomedical narrative text: an unsupervised method” *J Biomed Inform*. **34 (4)** (2001)