Method used for BioCreAtIvE Task 1A

Harald Kirsch*, Dietrich Rebholz†

February 27, 2004

We present a phrase pattern based approach to task 1A. Initially 4 phrase patterns mentioning protein names[1] in natural language text are applied to Medline and patent text to extract the protein names. A second list of alphanumeric protein names was generated from HuGo, IPI and Swiss-Prot. Both lists were integrated into more complex phrase patterns to fulfill task 1A. Precision of 67% and recall of 68% indicate that 2/3 of the annotation examples follow basic language concepts which can be easily modelled with patterns.

1 Introduction

Information extraction (IE) like Task 1A can be tackled mainly in two ways: (1) with the help of hand-crafted phrase patterns, syntactical rules or frames including syntax information [1, 2, 3] or (2) with the help of machine learning techniques [4, 5]. In both cases dictionaries and stop lists are applied to classify encountered terminology, and the use of a tagger integrates syntactical information and disambiguation of unspecific English terminology.

Pattern based methods provide interfaces to shape the patterns according to the investigator's demands and therefore do not depend on a prepared set of annotated data. In the case of complex sentence structure and of highly ambiguous terminology the patterns tend to be complex as well.

The method used by our group relies on hand crafted syntax patterns, protein names found in public databases and a small list of stop words to remove the most embarassing false positives. The following computational steps are run:

1. An elaborate analysis is applied to many publicly available databases to generate a dictionary of protein names.
2. Four different high precision/low recall patterns are applied to the 12M abstracts of Medline and 300000 patent applications from the biomedical field to detect protein names.
3. The resulting names are tagged as `NEWGENE` in the BioCreAtIvE sentences.
4. A set of patterns is applied to extend the names with prefixes like human or wild-type as well as with suffixes like subunit or beta-3.

*European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, kirsch@ebi.ac.uk

†dto., rebholz@ebi.ac.uk

[1]The term *protein name* should be understood to also include gene names.

2 Name Mining in Medline Abstracts, Patents and Biological Databases

Many uses of a protein name require the reader to know that it is a protein name in the same way he knows that the word *dog* denotes an animal. For other uses the reader will derive it from a context of sometimes several sentences or even paragraphs. Some uses, however, explicitly state the fact that the name denotes a protein. Examples are

1. *The AZ2 protein was ...* (PMID 10580148),
2. *The gene ATP6H, ...* (PMID 11471056),
3. *... binding domain of LP ...* (PMID 9560442),
4. *PMP22 is the crucial gene ...* (PMID 7628084).

Our idea was to exploit the fact that relevant protein names will at one point be introduced as such in one abstract of MEDLINE. Consequently we used four patterns generalizing the above examples to scan MEDLINE as well as 300000 patent applications in the biomedical domain[2] for protein names. The patterns are as follows:

1. `the` X `protein`
2. `the protein` X
3. T `domain of` BNP
4. ⊥ BNP `is a protein`

The X denotes a single token the syntax of which was even restricted to something which is most likely not a normal English word. BNP, for *basic noun phrase*, denotes a noun phrase comprised only of adjectives and nouns in a certain order. The T represents a set of triggers which make the *domain of* more likely to refer to a protein. Finally ⊥ denotes an anchor, like the beginning of the sentence, which was used to suppress spurious matches. In addition, *the*, *is*, *a* and *protein* have to be understood as placeholders for sets of tokens we allow at these positions.

Pattern 1: "the X protein"
For the word "protein" in fact the regular expression

```
[Gg]enes?|[Pp]roteins?|[a-z]*expressions?
|[Mm][Rr][Nn][Aa]s?
```

was used. The "the" was allowed to be any determiner, conjunction, preposition, verb or adverb. An example match is *induces Serrate-1 expression*[3].

Pattern 2: "the protein X"
The regular expression used instead of "protein" was now

```
[Gg]enes?|[Pp]roteins?|[Mm][Rr][Nn][Aa]s?
|[Tt]ranscription[a-z]* [Ff]actors?
|[Pp]rotein [Kk]inases?.
```

[2] this preselected set of patent applications is available in the EBI internally
[3] PMID: 9108364

Allowing "expression" is this pattern does not make much sense, however "transcription factor" and "protein kinase are quite common, as in the example *via transcription factor pit-1*[4].

Pattern 3: "T domain of BNP"
Originally, "domain of" was also allowed to be "expression of", but early experiments showed that this pattern only has roughly 50% precision. The T codes a regular expression which used to increase the probability that "domain of" is used in a biological context. It is

```
binding|terminal|cellular|death|cytoplasmic|catalytic
|homology|globular|[A-Z0-9]*[A-Z]+[A-Z0-9]*
```

The last branch allows any sequence of uppercase characters which turned out to be quite precisely indicating a biological context, as in the example *ABC domain of vitamin D receptor*[5]. As mentioned above, BNP denotes a basic noun phrase coded as

$$ADJ\ NOUN^{+}\ SUFFIX^{*}$$

ADJ and *NOUN* are words reported with this part-of-speach by a tagger[6]. The *SUFFIX* is the regular expression

```
[0-9]+|alpha|beta|gamma|kappa
|[A-Z]|II|III|IV|VI|VII|VIII
```

Pattern 4: "⊥ BNP is a protein"
We used the beginning of a sentence, a punctuation mark like a comma and the words "that" and "because" as an anchor in front of the basic noun phrase to suppress spurious matches. Coordinating words like "and" and "or" were explicitely left out as they may as well be part of a noun phrase. For "protein" we used the same indicators as with pattern 2.

For patterns 1 and 2 it has to be noted that X had to conform to a restrictive pattern. We formed regular expressions which made sure that X contained at least one uppercase letter not as the first character, contained no vowels or contained the regular expresson `[0-9/:.]`.

A little statistic about the extracted protein names is given in table 1. In most interesting column *names* we see that we could extract 128k distinct names from medline abstracts and 96k names from the patents. The two numbers cannot be meaningfully totalled, since many names appear in the patents as well as in MEDLINE. A one line shell script reveals that in fact there are 127k different names.

We estimated the precision by randomly choosing 100 names found in MEDLINE by each of the patterns. Table 2 shows that patterns 1, 3 and 4 offer sufficiently high precision. In fact it is hard to believe how they can still go wrong, but for example in PMID 11856371 pattern 1 picked up the phrase *...that PRMT1/BTG proteins...* which actually denotes two proteins.

Slightly disappointing was the performance of *domain of* (pattern 3). Too often it picks up general concepts instead of specific names like in *...cytoplasmic domain of recombinant transmembrane proteins...*[7].

In addition to the list of protein names generated as described above, protein names which consist only of alphanumeric signs were extracted from the protein name field of the following

[4]PMID: 1775132
[5]PMID: 11476956
[6]developed at CIS *Centrum für Informations- und Sprachverarbeitung* at the university of Munich
[7]PMID 11380458

source	pattern	documents	names	a-strings
MEDLINE	1	370k	95k	53k
	2	43k	19k	15k
	3	16k	9k	8k
	4	7k	5k	5k
	total	<436k	<128k	<81k
patents	1	233k	76k	44k
	2	26k	11k	9k
	3	15k	7k	6k
	4	4k	2k	2k
	total	<278k	<96k	<61k

Table 1: Some statistics about protein names found. The column *documents* denotes the number of documents with at least one match. The *names* column lists the number of unique names, while *a-strings* counts unique strings left when only keeping `[A-Za-z0-9]` in a name and converting to lowercase. The totals are upper bounds since patterns 3 and 4 may find the same names again as 1 and 2.

pattern	precision
1	95%
2	96%
3	87%
4	98%

Table 2: Precision of the four patterns measured on 100 randomly choosen distinct names found in MEDLINE.

data sources: Hugo, Ipi, SwissProt. Apart from an obvious stop list to delete malicious examples like *WAS* and *NOT*, several steps of filtering were applied to get terms which have a chance at all to be found in MEDLINE.

3 Tagging and Extending Protein Names in the BioCreAtIvE sentences

The collection of names described in the previous two sections were collected independently of BIOCREATIVE. Then BIOCREATIVE came along being a nice test case for the extracted proteins. First, all names of the collection were tagged as `NEWGENE` in the BIOCREATIVE corpus. Then they were extended by applying what we called *headers*, *pretrailers* and *trailers*. Examples are *human alpha-II* and *binding site* respectively. The full lists are as follows:

header: human, mouse, serum, rat, yeast, wild type, virus, viral, mammalian, murine, drosophila, mutant, cat, bovine, saccharomyces cerevisiae, chicken, xenopus, escherichia coli, mitochondrial, eurkaryotic, hiv-1, cellular, bacterial, extracellular, rabbit, porcine

pretrailer: alpha`[A-Za-z0-9-]*`, beta`[A-Za-z0-9-]*`, gamma`[A-Za-z0-9-]*`, delta`[A-Za-z0-9]*`, kappa`[A-Za-z0-9-]*`, I, II, III, IV, V, VI, VII, VIII, `[A-Z]`, `[0-9]+`

trailer: monomer, codon, region, exon, orf, cDNA, reporter gene, antibody, complex, gene product, mrna, oligomer, chemokine, subunit, peptide, message, transactivator, homolog, binding site, enhancer, element, allele, isoform, intron, promoter, operon, mRNA, mutant,

protein, gene, protein kinase, kinase, structure, family, polypeptide, peptide, motif, dimer, domain

Defining

gene = `NEWGENE(/NEWGENE)?`

as a shortcut, we used the the following regular expressions to perform the extension:

header gene trailer*
header gene pretrailer trailer**
*header gene pretrailer? trailer**

4 Result and Discussion

In our approach we used protein names generated from Medline and from biological databases to identify proteins and genes mentioned in the test data. The performance in terms of precision and recall is unsatisfactory. Low recall was due to the fact that only part of the match was identified from the patterns. Two reasons accounted to this fact: (1) insufficient complexity of the patterns which was deliberately taken into account, and (2) inconsistencies in the annotation of the test data. Improvements to our method could result from postprocessing of the extracted match to extend and fit the encountered match to its environment.

On the other side, our approach shows that a large portion of the phrases mentioning proteins follows basic rules, e.g. *header* gene trailer*, and can be well modeled with a small number of patterns. Increasing the recall requires an increase in patterns which induces an increase in overhead and which can be interpreted as another instance of Zipf's law.

References

[1] D. Rebholz-Schuhmann, S. Marcel, S. Albert, R. Tolle, G. Casari and H. Kirsch (2004): *Automatic extraction of mutations from Medline and cross-validation with OMIM.* Nucleic Acids Research, Vol. 32(1), pp 135-142

[2] R. Gaizauskas, G. Demetriou, P.J. Artymiuk and P. Willett (2003): *Protein Structures and Information Extraction from Biological Texts: The PASTA System.* Bioinformatics, Vol 19(1), pp 135–143

[3] J.M. Temkin and M.R. Gilder (2003): *Extraction of protein interaction information from unstructured text using a context-free grammar.* Bioinformatics, Vol 19(16), pp 2046–2053

[4] V. Hatzivassiloglou, P.A. Duboue and A. Rzhetsky (2001): *Disambiguating proteins, genes, and RNA in text: a machine learning approach.* Bioinformatics, Vol 17 (Suppl 1), pp S97–S106.

[5] B.J. Stapley, L.A. Kelley and M.J.E. Sternberg (2002): *Predicting the sub-cellular location of proteins from text using support vector machines.* PSB.