

# 國立交通大學

資訊科學系

碩士論文

生物語料中蛋白質名稱之自動辨識

Automatic Protein Entities Recognition from PubMed Corpus

研究生：施並格

指導教授：梁 婷 教授

中華民國九十三年六月


# 生物語料中蛋白質名稱之自動辨識

研究生：施並格

指導教授：梁婷 博士

國立交通大學資訊科學研究所

## 摘要



一般而言，專有名詞的語意辨識是建立專業知識庫自動化過程的一項基本且重要的工作。此種語意辨識方法可以分為規則式與統計式兩種。在本篇論文，我們分別檢視這兩種方法在生物領域上的效果。規則式的方法以核心詞、功能詞、及已定義詞為基礎，配合詞性標記來辨識蛋白質名稱，再利用六條規則來提升系統的效能，實驗針對 GENIA 及 SwissProt Reference 語料作測試，規則式的系統分別可以達到 52%、51% 的 F 分數。統計式的方法利用萃取出的內部特徵、外部特徵、及全域特徵，以簡潔的馬可夫模型為基礎，並配合 back-off 的機率模型以解決資料稀疏的問題，實驗同樣針對 GENIA 及 SwissProt Reference 語料作測試，統計式的系統皆可以達到 77% 的 F 分數。除此之外，我們亦使用歸納的經驗法則來發掘出在變化詞中的省略詞彙，實驗結果可得到 89% 的求全率與 69% 的求準率。

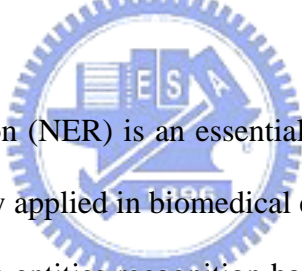
# Automatic Protein Entities Recognition from PubMed Corpus

Student: Ping-Ke Shih

Advisor: Tyne Liang

Institute of Computer and Information Science  
National Chiao Tung University

## Abstract



Named Entity Recognition (NER) is an essential task of knowledge acquisition. Recently NER has been widely applied in biomedical entities extraction. In this thesis, we proposed automatic protein entities recognition based on rule-based and statistical approaches. Rule-based approach relies on core terms, function terms, predefined terms and Part-of-Speech tags. Then six rules are applied to boost performance. The experiments with GENIA and SwissProt Reference corpus, rule-based approach can yield 52% and 51% F-score respectively. Statistical approach is based on concise Hidden Markov Model, and back-off models are conducted to overcome data sparseness problem. We use not only internal, external, global features but also the result of rule-based approach to identify protein entities. Statistical approach can yield 77% F-score in both GENIA and SwissProt Reference corpus. Besides, we use heuristic rules to mine hiding named entities and expand them out of coordination variants. Term variants resolution system can yield 89% recall and 69% precision.

# ACKNOWLEDGEMENTS

Thank my advisor Dr. Tyne Liang for her encouragement and teaching during the two years, and then this thesis can be accomplished. Besides, she gives me a guide to make plans for the future.

Also, I thank the members of information retrieval laboratory including Chien-Pang Wang, Dian-Song Wu, Chih-Chien Chao, Yu-Hsiang Lin, Lan-Chi Lin, Yi-Li Chen, Yi-Chia Wang, and Hsiao-Ju Shih. They provide suggestions and comments to assist in writing. Special thanks to my girlfriend Mei-Hua Wang for her support, concern and patience, and then we stroll along colorful and happy life forever.

Finally, I thank my family for support and encouragement in the period of school life. I will dedicate this achievement to my parents who are the greatest farmers on earth.



# CONTENTS

<b>ABSTRACT (in Chinese)</b> .....	<b>i</b>
<b>ABSTRACT (in English)</b> .....	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b> .....	<b>iii</b>
<b>CONTENTS</b> .....	<b>iv</b>
<b>LIST OF TABLES</b> .....	<b>vi</b>
<b>LIST OF FIGURES</b> .....	<b>vii</b>
<b>Chapter 1 Introduction</b> .....	<b>1</b>
<b>Chapter 2 Related Works</b> .....	<b>4</b>
<b>2.1. Biomedical Resources</b> .....	<b>4</b>
2.1.1. PubMed and MEDLINE .....	4
2.1.2. Swiss-Prot .....	5
2.1.3. GENIA corpus.....	6
<b>2.2. Named Entity Extraction</b> .....	<b>9</b>
2.2.1. Rule-based Approach .....	9
2.2.2. Statistical Approach .....	10
2.2.3. Hybrid Approach.....	12
<b>Chapter 3 Preprocessing</b> .....	<b>15</b>
<b>3.1. Corpus Preparation</b> .....	<b>15</b>
<b>3.2. Text Preprocessing</b> .....	<b>17</b>
3.2.1. Segmentation and Tokenization .....	17
3.2.2. Part-of-Speech Tagger.....	18
<b>3.3. Term Variants Resolution</b> .....	<b>19</b>
<b>3.4. Noisy Filtering</b> .....	<b>22</b>
<b>Chapter 4 Named Entity Extraction and its Results</b> .....	<b>24</b>
<b>4.1. Rule-based Named Entity Extraction</b> .....	<b>24</b>
<b>4.2. Statistical and Hybrid Named Entity Extraction</b> .....	<b>31</b>
4.2.1. Features Extraction .....	31

4.2.2.	HMM Modeling .....	34
4.2.3.	Back-off Modeling .....	36
<b>4.3.</b>	<b>Systems Comparison.....</b>	<b>38</b>
4.3.1.	Experiment with SRC .....	38
4.3.2.	Experiment with GENIA Corpus .....	39
<b>Chapter 5</b>	<b>Conclusions and Future Works .....</b>	<b>41</b>
<b>References.....</b>		<b>43</b>
<b>Appendix A.....</b>		<b>47</b>
<b>Appendix B.....</b>		<b>48</b>
<b>Appendix C.....</b>		<b>50</b>



# LIST OF TABLES

Table 2-1: The statistics of GENIA corpus version 3.02p. ....	8
Table 3-1: The basic statistics in SRC. ....	16
Table 3-2: The result of 10-fold cross validation in GENIA 3.02p.....	19
Table 3-3: Notation of coordination variants. ....	20
Table 3-4: Original pattern, expanded pattern, and examples. ....	20
Table 3-5: The statistics of coordination variants on training and testing data.....	21
Table 3-6: The statistics of cluster and term number. ....	21
Table 3-7: Accuracy of coordination variants identification in GENIA 3.02p. ....	22
Table 3-8: The number of protein names after term variants resolution and the number of resolved instances. ....	22
Table 3-9: The statistic of the number of tokens in the defined regions.....	23
Table 4-1: The three predefined types.....	25
Table 4-2: The effect of term-based stage (step 1 to 3).....	27
Table 4-3: The examples of the patterns '<math>\langle T_{-2}, T_{-1}, \#, T_1, T_2 \rangle</math>'. ....	28
Table 4-4: The examples of the 4 bag-of-word collected from the surroundings of protein entities.....	28
Table 4-5: The list of significant clue tokens. ....	29
Table 4-6: The rule-based result in SRC and GENIA.....	30
Table 4-7: The intermediate results of rule-based approach. ....	31
Table 4-8: The internal features, and their descriptions and examples.....	32
Table 4-9: The top 20 examples of prefix and suffix strings. ....	32
Table 4-10: A sentence and its corresponding external features. (Where 'R_BIO' is the result of rule-based named entity extraction.).....	33
Table 4-11: The significant nouns according to chi-square estimation.....	34
Table 4-12: The comparison between HMM models.....	37
Table 4-13: The effects of features in concise HMM. ....	38
Table 4-14: Comparison between rule-based systems in SRC. ....	38
Table 4-15: Comparison with other systems in GENIA version 3.x.....	39
Table 4-16: Comparison with other systems in GENIA version 1.1.....	40
Table 4-17: Comparison with rule-based and hybrid approaches.....	40

# LIST OF FIGURES

Figure 1-1: The amount of PubMed citations.....	3
Figure 1-2: The amount of SwissProt entities.....	3
Figure 2-1: A PubMed reference coded in XML format. [ <a href="http://www.ncbi.nlm.nih.gov">http://www.ncbi.nlm.nih.gov</a> ] .....	5
Figure 2-2: An example of annotated abstract in GPML format in GENIA corpus version 3.02.....	7
Figure 2-3: The length distribution of protein entities in GENIA corpus version 3.02p. .....	8
Figure 3-1: The length distribution of protein name in SRC and GENIA.....	16





# Chapter 1

## Introduction

With the rapid growth of biomedical research, huge amounts of biomedical resources are available. For example, the amount of biomedical citations available by PubMed increases 68.95% in recent ten years (Figure 1-1). Because there are huge amounts of biomedical texts, automatic knowledge acquisition is very important. Besides, it is impossible to extract knowledge by human experts, so natural language process (NLP) and machine learning techniques will be applied.

To discover knowledge, named entity recognition (NER) is one of the essential tasks, because named entities have special significance in texts. In Message Understanding Conference (MUC), Named Entities (NE) are defined as proper names and quantities of interest. Person, organization, and location names are marked as well as dates, times, percentages, and monetary amounts. The best result of named entity task can yield 94% and 97% in MUC-7 and MUC-6 respectively ([http://www.itl.nist.gov/iaui/894.02/related\\_projects/muc/proceedings/muc\\_7\\_proceedings/overview.html](http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_proceedings/overview.html)), but the best result is 66.5% F-score in GENIA 3.0 [Lee et al., '03]. In biomedical domain, there are many problems such as open vocabulary, synonyms and boundaries. One reason is that biomedical entities are continually increasing. For example, the number of entries in SwissProt, which is a protein knowledge base, increases 277.36% in recent ten years (Figure 1-2). In view of

biology, protein plays important role in the discovery of life. Because proteins are required for the structure, function, and regulation of the body's cells, tissues, and organs, and each protein has unique functions. Therefore, we aim at protein entities recognition.

There are three NER methods namely rule-based, statistical and hybrid methods. Generally, rule-based systems use terms and rules (e.g. heuristic rules and decision tree rules) to produce candidates, and lexical analysis is applied to judge candidates. Two famous systems developed by rule-based approaches are KeX [Fukuda et al., '98] and Yapex [Olsson et al., '02]. Statistical approaches are based on Hidden Markov Model (HMM), Support Vector Model (SVM), Maximum Entropy (ME), and Naïve Bayes. Out of these approaches, HMM and SVM outperform others and yield about 70% F-score in GENIA corpus version 3.0 (an annotated corpus in biomedical domain). The identification phase of hybrid methods rely on parsers or chunkers. The kernel classification mechanism of hybrid methods is based on statistical models. Statistical approaches are scalable and portable, yet its performance depends on high-quality corpus.

In this thesis, both rule-based and statistical approaches are investigated to extract protein entities. Statistical approach is based on Hidden Markov Model, and a back-off model is conducted to overcome data sparseness problem. Proposed approaches are applied to two corpora GENIA 3.02p and the assembled SRC (SwissProt\_Ref Corpus). The result shows that our approaches can yield 77% F-score in both corpora, i.e. our approaches can adapt to different corpus successfully. Moreover, back-off models can boost precision, because severe probability models can be applied and then the relaxed ones. The experimental result show that our system can yield 75.22% recall, 78.12% precision and 76.64% F-score in GENIA corpus 3.02p.

Most of NER extractors deal with proper names only, and common phenomena appearing in written texts like pronominal anaphora, definite anaphora and term variants are not concerned. In this thesis, we solve coordination variants, and our resolver obtains 89.09% recall, 56.11% precision and 68.85% F-score.

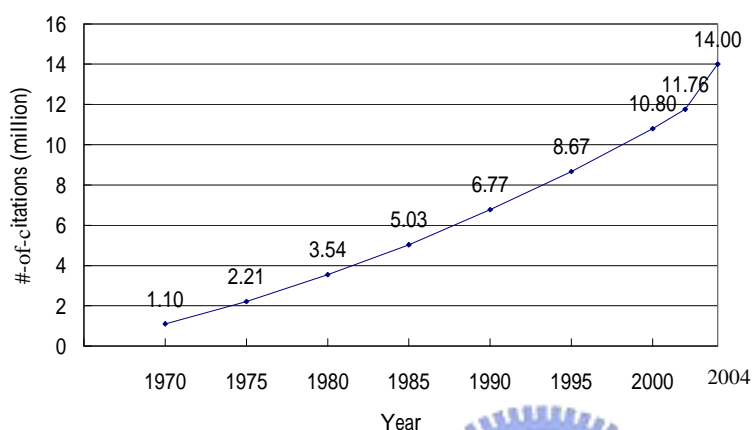


Figure 1-1: The amount of PubMed citations.

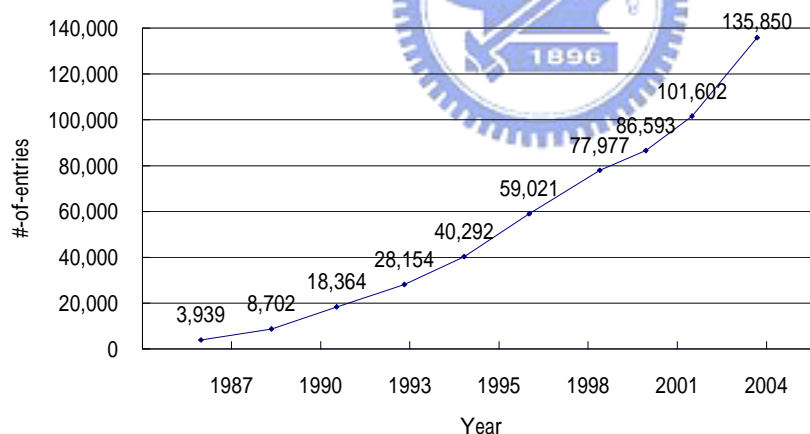


Figure 1-2: The amount of SwissProt entries.

The organization of this thesis is as follows. Chapter 2 describes the related works and useful biomedical resources. Chapter 3 describes our assembled corpus and its preprocessing. Chapter 4 presents the proposed extractors and corresponding experimental results. Chapter 5 gives the conclusion and future works.

# Chapter 2

## Related Works

In this chapter, the resources related to the proposed extractor including Swiss-Prot, PubMed citations and GENIA corpus, and the NER techniques proposed in recent literature will be addressed. Swiss-Prot is an annotated protein knowledge base. PubMed is a search engine for accessing biomedical literature. GENIA is developed to extract useful information automatically.



### 2.1. Biomedical Resources

#### 2.1.1. PubMed and MEDLINE

PubMed is one of the services provided by Entrez which is a text-based search and retrieval system used at NCBI (National Center for Biotechnology Information) (<http://www.ncbi.nlm.nih.gov/>). PubMed was designed to provide access to citations from biomedical literature.

MEDLINE citations and abstracts are available as the primary component of PubMed's database. MEDLINE contains bibliographic citations and author abstracts from more than 4,600 biomedical journals published in the United States and 70 other countries. The database contains over 12 million citations dating back to the

mid-1960's.

PubMed provides many formats, including ASN.1, GEN, XML, and FASTA.

Figure 2-1 is an example of PubMed reference coded in XML.

```
<?xml version="1.0" ?>
<!DOCTYPE PubmedArticleSet (View Source for full doctype...)>
- <PubmedArticleSet>
- <PubmedArticle>
  - <MedlineCitation Owner="NLM" Status="Completed">
    <PMID>91354</PMID>
    + <DateCreated>
    + <DateCompleted>
    + <DateRevised>
    + <Article>
    + <MedlineJournalInfo>
    + <ChemicalList>
      <CitationSubset>IM</CitationSubset>
    + <MeshHeadingList>
    </MedlineCitation>
  - <PubmedData>
    + <History>
      <PublicationStatus>ppublish</PublicationStatus>
    + <ArticleIdList>
    </PubmedData>
  </PubmedArticle>
</PubmedArticleSet>
```

Figure 2-1: A PubMed reference coded in XML format. [<http://www.ncbi.nlm.nih.gov>]

## 2.1.2. Swiss-Prot

Swiss-Prot Protein knowledge base is an annotated protein sequence database [Boeckmann et al., '03]. It was established in 1986 and maintained collaboratively, since 1987, by the group of Amos Bairoch first at the Department of Medical Biochemistry of the University of Geneva and now at the Swiss Institute of

Bioinformatics (SIB) and the EMBL Data Library (now the EMBL Outstation - The European Bioinformatics Institute (EBI)). The knowledge base is a curated protein sequence database that provides a high level of annotation, a minimal level of redundancy and high level of integration with other databases.

Swiss-Prot version 42.0 (October 2003) is released in plain text format, and its size is 465MB. This release contains 135,850 entries, and categories of each entry include entry information, name and original of the protein, references, comments, copyright, cross-references, keywords, features, and sequence information. The name and original of the protein include protein names and their synonyms, and the references of the protein.

There are 135,850 entries and each entry contains 2.54 synonyms in average. There are 13,048 entries without PubMed reference. Hence, we have 222,522 PubMed reference and there are 1.81 PubMed reference per entries. Out of 222,522 PubMed references, there are 88,437 unique references. Out of these references, we collect those ones containing titles and abstracts. Then we download the 82,740 abstracts according to their reference IDs, namely PMIDs.

### 2.1.3. GENIA corpus

The GENIA corpus is annotated by Ohta et al. [‘02] in GENIA project at the Tsujii laboratory of the University of Tokyo. The abstracts of this corpus are taken from MEDLINE database by using the MeSH query terms, ‘*Human*’, ‘*Blood Cells*’, and ‘*Transcription Factors*’.

GENIA corpus is encoded in GENIA Project Markup Language (GPML), a document type definition (DTD) of XML language that consists of definitions of structural elements, linguistic elements, and resource elements. Each record contains

an 8-digits MEDLINE ID. Figure 2-2 shows an example of the annotated MEDLINE abstract in GPML format.

```

<article>
<articleinfo>
  <bibliomisc>MEDLINE:95343554</bibliomisc>
</articleinfo>
<title>
  <sentence>
    <cons lex="E1A_gene_expression" sem="G#other_name"><cons lex="E1A_gene" sem="G#DNA_domain_or_region">E1A
    gene</cons> expression</cons> induces susceptibility to killing by <cons lex="NK_cell" sem="G#cell_type">NK cells</cons>
    following immortalization but not <cons lex="adenovirus_infection" sem="G#other_name"><cons lex="adenovirus"
    sem="G#virus">adenovirus</cons> infection</cons> of <cons lex="human_cell" sem="G#cell_type">human cells</cons>.
  </sentence>
</title>
<abstract>
  <sentence>
    <cons lex="adenovirus_(Ad)_infection" sem="G#other_name"><cons lex="adenovirus" sem="G#virus">Adenovirus</cons> (Ad)
    infection</cons> and <cons lex="E1A_transfection" sem="G#other_name"><cons lex="E1A"
    sem="G#protein_molecule">E1A</cons> transfection</cons> were used to model changes in susceptibility to <cons
    lex="NK_cell_killing" sem="G#other_name">NK cell killing</cons> caused by transient vs stable <cons lex="E1A_expression"
    sem="G#other_name"><cons lex="E1A" sem="G#protein_molecule">E1A</cons> expression</cons> in <cons lex="human_cell"
    sem="G#cell_type">human cells</cons>.
  </sentence>
  ...
</abstract>
</article>

```

Figure 2-2: An example of annotated abstract in GPML format in GENIA corpus version 3.02.

According to the ontology defined in GENIA project, there are six semantic types related to protein entities, namely *protein\_N/A*, *protein\_complex*, *protein\_domain\_or\_region*, *protein\_molecule*, *protein\_substructure*, and *protein\_subunit*. Table 2-1 shows that protein entities occupy 42.50% of biomedical named entities. Average length of protein entities is 1.79, and the length distribution is shown in Figure 2-3.

	Count	Average
Abstract	1,999	
Sentence	18,572	9.29 (s/a)
Token	490,469	245.36 (t/a)
		26.41 (t/s)
Named Entity	76,526	38.28 (ne/a)
		4.12 (ne/s)
Protein Entities	32,525	16.27 (pe/a)
		1.75 (pe/s)

Table 2-1: The statistics of GENIA corpus version 3.02p.

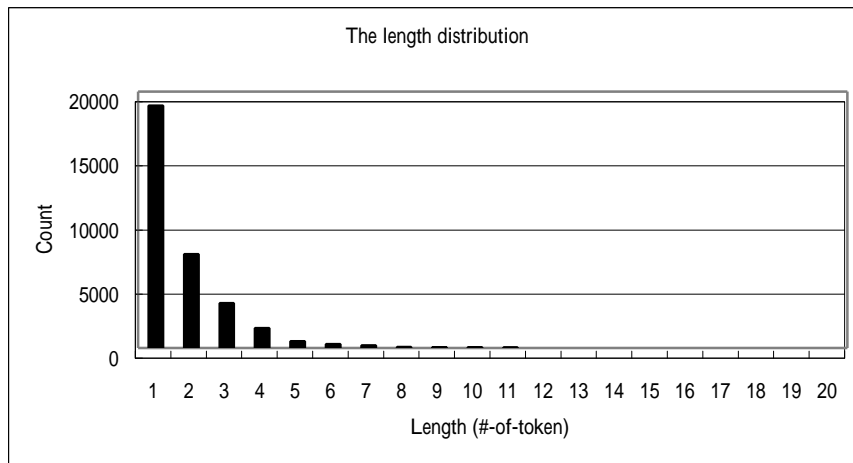


Figure 2-3: The length distribution of protein entities in GENIA corpus version 3.02p.

(The maximum length is 20.)



## 2.2. Named Entity Extraction

### 2.2.1. Rule-based Approach

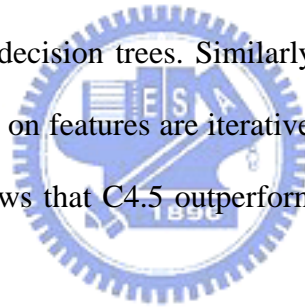
There are many rule-based approaches presented in recent literatures [Fukuda et al., '98; Gaizauskas et al., '00; Hatzivassiloglou et al., '01; Olsson et al., '02; Narayanaswamy et al., '03; Hanisch et al., '03]. Sufficient domain knowledge is required to construct the rules, and this method is lack of portability and scalability.

There are two famous rule-based named entity systems: One is KeX developed by Fukuda et al. ['98], and another one is Yapex developed by Olsson et al. ['02]. Fukuda et al. ['98] define core terms (e.g. 'transcription factor **E2F-1**') to provide the major information and treated them the kernel part of the name entity. Function terms (e.g. '**transcription factor** E2F-1') are used to describe the function and characteristic of a material name. Protein names are composed of one or more words with upper case letters, numerical letters, or non-alphabetical letters. Then rules were applied to rebuild core blocks (noun-phrases without conjunction and preposition), noun phrases and dependencies with core blocks. Finally, Fukuda's system can extract 95% of protein entities in 80 abstracts retrieved from MEDLINE. Olsson et al. ['02] use core and function terms to judge candidates (noun phrases). For example, a candidate has to contain a core term. Besides, regular expression patterns of suffixes are applied to filter non-protein entities (e.g. suffixes 'amic' and 'amide'). Narayanaswamy et al. ['03] proposed help terms (e.g. '**homolog** of TAF (II) 30<sub>protein</sub>') that provided clues about target classes but not considered part of name entities. Core, function and help terms were applied to classify six classes of named entities (protein/gene, protein/gene parts, chemical, chemical parts, source, and general classes), and this system can yield

91.90% F-score in 55 MEDLINE abstracts.

Hanisch et al [‘03] used six token classes namely *modifier*, *non-descriptive*, *specifier*, *common*, *delimiter* and *standard* classes. *Modifier* and *non-descriptive* classes were similar to function term. *Specifier* class was numbers and Greek letters. *Delimiter* class was separator tokens such as ‘(’ and ‘)’. Usually, the capitalization is insignificant except some gene entities (e.g. WAS and KILLER), so the search of *common words* is case-sensitive. All tokens not explicitly classified are placed in *standard* class. This system achieved 95% recall and 90% precision.

Hatzivassiloglou et al. [‘01] used three learning techniques to construct NER. The techniques are Naïve Bayesian learning, C4.5 and RIPPER. Naïve Bayesian learning aims to assign probabilities corresponding to target classes. C4.5 is a particular implementation of decision trees. Similarly, RIPPER is based on decision trees, but rules involving tests on features are iteratively constructed. The comparison with the three techniques shows that C4.5 outperforms Naïve Bayesian learning and RIPPER.



## 2.2.2. Statistical Approach

Famous statistical models have been applied in biomedical named entity extraction such as Hidden Markov Model, Maximum Entropy, Support Vector Machine, and Naïve Bayes [Nobata et al., ‘99; Collier et al., ‘00; Kazama et al., ‘01; Kazama et al., ‘02; Chieu and Ng, ‘03; Shen et al., ‘03; Takeuchi and Collier, ‘03; Tsuruoka and Tsujii, ‘03; Yamamoto et al., ‘03].

Most statistical methods are sensitive to features selection except SVM. In order to determine features effect, Kazama et al. [‘02] examined many features, and found that *Left Word Cache* (biomedical terms appear in left side), *Right Word Cache*

(biomedical terms appear in right side), *Preceding Class* and *Suffix* play positive effect. Lee et al. [‘03] presented two-phase recognition method, and found that features have different contribution in different phases. In identification phase, positive features are *Word*, *Part-of-Speech*, *Suffix*, and *Prefix*. To classify entities, predefined *Functional Words*, *Word List*, *Nouns*, and *Verbs* are used. Besides, *Surface*, *Morphological*, *Contextual* features are widely used [Nobata et al., ‘99; Collier et al., ‘00; Takeuchi and Collier, ‘02; ‘03; Shen et al., ‘03; Tsuruoka and Tsujii et al., ‘03; Yamamoto et al., ‘03].

The learning approach of Hidden Markov Model (HMM) is generative, i.e. it uses positive examples to build a model of named entity classes and then evaluates unknown entity to assign an appropriate class. Collier et al. [‘00] used HMM interpolation model to overcome the problem of data sparseness, but the model was slightly complex due to manual arguments. With surface word features, the HMM classifier can achieve a 72.8% F-score in 100 MEDLINE abstracts. In addition to Collier’s word feature, Shen et al. [‘03] adopted morphological feature, Part-of-Speech features, and semantic trigger features, and their system can achieve 66.1% F-score in GENIA corpus version 3.0.

Support Vector Machine (SVM) is a discriminative approach and uses positive and negative examples to learn the distinction between the two classes. Because SVM is a binary classifier, many modified versions are proposed to solve multi-class problem [Kazama et al., ‘02]. There are two popular methods, namely one-vs-rest and pairwise. The one-vs-rest method constructs  $K$  ( $K$  denotes the number of target classes) binary SVMs, each of which determines whether the testing data should be classified as class  $i$  or as the other classes. The output is the class with the maximum margin, and one-vs-rest takes time in  $K \times O_{SVM}(L)$  ( $L$  denotes the number of training sample, and  $O_{SVM}$  represents super-linear complexity of SVM). The pairwise method

constructs  $K(K-1)/2$  binary SVMs, each of which determines whether the testing data should be classified as class  $i$  or class  $j$ . Each binary SVM has one vote, and the output is the class with the maximum votes. Then pairwise method takes time in  $K(K-1)/2 \times O_{SVM}(2L/K)$ . Because SVM training is a quadratic optimization program, its cost is super-linear to the size of training samples. Consider time complexity, pairwise method is better than one-vs-rest method [Kazama et al., '02; Takeuchi and Collier, '03; Yamamoto et al., '03]. In our studies, the performances of SVM and HMM are almost the same.

Tsuruoka and Tsujii [03] used an approximate string searching technique to produce candidates, and used the traditional Naïve Bayes formula and a binary feature vector to filter candidates. In GENIA 3.0, the F-score is improved from 57.6% (without filter) to 69.5%.

The entity recognitions based on maximum entropy are discussed in [Nobata et al., '99; Kazama et al., '01; Chieu and Ng, '03]. The kernel of maximum entropy is to maximize the likelihood of the training data by using numerical optimization methods such as Generalized Iterative Scaling (GIS). This is an iterative method that improves the estimation of parameters at each iteration. The tagging procedure is formulated as a searching question to find a sequence with the maximum probability.

### 2.2.3. Hybrid Approach

There are some systems built only for identifying a certain entity class, and these systems consist of two phases. One is to find candidates, and the other is to validate the candidates. In first phase, a dictionary-based method is applied to the systems to identify typical words or known terms. For example, Proux et al [98] used lexical rules, Wilbur et al. [99] used rule-based segmentation, and Tsuruoka and

Tsujii [‘03] used approximate string searching. In order to judge these candidates, Hidden Markov Models [Proux et al., ‘98], and Naïve Bayes [Wilbur et al., ‘99; Tsuruoka and Tsujii, ‘03] are used.

Some systems were built for entity classification only. Torill et al. [‘03] used function terms (e.g. *protein* and *receptor*), suffixes and string similarity to classify. The classifier will assign an appropriate class, if the last words of entities appear in function terms. Otherwise, the classifier tried to classify according to the suffixes of the last words. If entities still cannot be classified, a score of string similarity is calculated to determine classes. Because this system needs not to care about identification, it yields 90% precision and 87% recall in GENIA corpus version 3.0.

A statistical named entity recognizer often use a central model, such as HMM and SVM. While most of these models are black boxes, it is difficult to adjust parameters. Therefore, we always choose a ‘good enough’ parameter, i.e., no one knows whether it is a global optimization or not. Shen et al. [‘03] proposed HMM named entity recognition, and they incorporate with abbreviation recognition and cascaded phenomena. It is clear that abbreviation and its full form must be the same class, i.e., if we know the full form, the abbreviation is also classified. Besides, the cascaded phenomena are clue to classify (e.g. ‘<Protein><DNA>kappa 3</DNA> binding factor</Protein>’). Basic types of cascaded NEs are found, and a set of cascaded patterns is learnt in training data. Then these patterns can be applied to testing data to correct the class.

	Fukuda '98	Gaizauskas '00	Hatzivassiloglou '01	Olsson '02	Narayanaswamy '03	Hanisch '03	Ours
Preprocess	Brill Tagger	No	Brill Tagger	Conexor Oy	No	No	Tagger
Dictionary	No	SWISS-PROT, CATH, SCOP	No	Swiss-Prot + Dynamic Dictionary	KeX results	HUGO	No
Abbreviation	No	No	No	No	Yes	No	No
Surface	Yes	No	No	Yes	Yes	No	No
POS	Yes	No	Yes	No	No	No	Yes
Suffix	No	Yes	No	No	No	No	No
Core Term	Yes	No	No	Yes	Yes	No	Yes
Function Term	Yes	No	No	Yes	Yes	Yes	Yes
Help Term	No	No	No	No	Yes	No	No
Other Term/Features			bag-of-word, biological terms distance	Specifier		Six Token Classes	Predefined (specipefier, Unit)
Chunker	Core, Function Terms	Grammer Rules	-	Min. NP	Core, Function Terms	Expansion, Pruning	Core, Function, Predefined Terms
Classifier	-	Terms Decomposition	RIPPER C4.5 Naïve Bayes	-	C-Term ( 3 types) F-Term ( 6 types )	-	-
Performance (F-score)	SGN: 93.6 SH3: 96.7	PASTA: 90.9 EMPATHIE: 75.9	85	67.1	91.9	-	50.7
Corpus	SGN, SH3 (MEDLINE)	PASTA, EMPATHIE	1374 from EMBO	48 MEDLINE 53 GENIA	55 MEDLINE	611 MEDLINE	GENIA 3.02p
# of Classes	1 (Protein)	PASTA: 13 EMPATHIE: 10	3	1 (Protein)	6	1 (Protein)	1 (Protein)

Table 2-2: A summary of rule-based approaches.

	Collier '00	Kazama '02	Takeuchi and Collier	Shen '03	Takeuchi and Collier	Tsuruoka and Tsujii '03	Lee '03	Ours
Preprocess	No	Tagger	No	Tagger	Tagger	No	Tagger	Tagger
Dictionary	No	Yes	No	No	No	Yes	Yes	No
POS	No	Yes	Yes	Yes	Yes	No	Yes	Yes
Surface	Yes	No	Yes	Yes	Yes	No	No	Yes
Prefix/suffix	No	Yes	No	Yes	No	No	Yes	Yes
Substring	No	Yes	No	No	No	No	No	No
(generalized) Cue Noun	No	No	No	Yes	Yes	No	Yes	Yes
(generalized) Cue Verb	No	No	No	No	No	No	Yes	No
Other Features								Output of Rule-based
Chunker	-	-	-	-	-	Approximate String Searching	SVM	-
Classifier	HMM	SVM	SVM	HMM	SVM	Naïve Bayes	SVM	HMM
Ident. Accu.	-	73.6	-	-	-	N/A	79.9	-
Class. Accu.	71.8	54.4	54.4	66.1	74	70.2	66.5	76.6
Corpus	BIO1	GENIA 1.0	BIO1	GENIA 3.0	BIO1	GENIA 3.0	GENIA 3.0p	GENIA 3.02p
# of Classes	10	6	10	23	10	1 (Protein)	22	1 (Protein)

Table 2-3: A summary of statistical approaches.

# Chapter 3

## Preprocessing

### 3.1. Corpus Preparation

The references in Swiss-Prot were assembled as ‘*SwissProt\_Ref Corpus*’ (‘*SRC*’ for short) to boost our protein entities extractor. To annotate protein entities existed in the assembled SRC is difficult for the following reasons:

- Pronoun: ‘IL-1 induces a rapid... . It also primes cells to...’, for example. The pronoun ‘It’ is the protein entity IL-2.
- Anaphora: ‘From a murine B-cell cDNA-library we have cloned a cDNA encoding the murine B-cell specific coactivator mBob1. The protein is the murine homologue to ...’, for example. ‘The protein’ refers to mBob1.
- Different forms: A protein name has variant spellings. For example, ‘IL 2’ and ‘IL-2’ are the same protein entity.
- Term variants: ‘Lyn and Jak2 tyrosine kinases’ refers to two entities ‘Lyn tyrosine kinase’ and ‘Jak2 tyrosine kinase’.

In this thesis, we addressed synonyms and term variants problems only. When we annotate SRC, the synonyms are dealt as follows:

Step 1. Tokens are split by space and hyphen.

Step 2. Each token is converted to lower case except to its initial character.

Step 3. If an entity exactly matches one of protein entities collected from SwissProt, the entity will be recognized.

Out of 82,740 PubMed references, we select 2,894 abstracts which contain at least six two-token protein entities. The basic statistics is shown in Table 3-1. Figure 3-1 shows the length distribution, and the average lengths of protein names are 1.81 and 1.79 in SRC and GENIA respectively. The protein entities whose lengths are shorter than three occupy 98% and 91% in SRC and GENIA respectively.

	Count	Average
Abstract	2,894	
Sentence	28,154	9.73(s/a)
Token	740,001	255.70(t/a) 26.28(t/s)
Protein Entity	31,977	11.05(pn/a) 1.14(pn/s)

Table 3-1: The basic statistics in SRC.

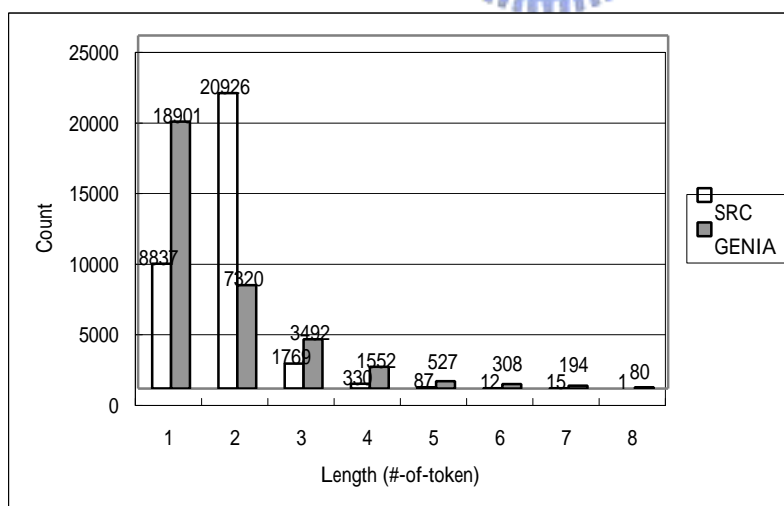


Figure 3-1: The length distribution of protein name in SRC and GENIA.



## 3.2. Text Preprocessing

### 3.2.1. Segmentation and Tokenization

*Sentence Splitter*, developed by Cognitive Computation Group at the Department of Computer Science, University of Illinois at Urbana-Champaign, is adopted to segment texts into sentences (<http://l2r.cs.uiuc.edu/~cogcomp/cc-software.html>). This program is written in Perl, and a list of honorifics is used to know the word ending with a full stop. The abbreviations are detected by regular expression.

While we take ‘( Genga , A. , Bianchi , L. , and Foury , F. ( 1986 ) J. Biol. Chem. 261 , 9328-9332 )’ as input of sentence splitter, it is split into three segments which are underlined. In order to solve this problem, we collect words appearing in a pair of parentheses or square brackets and their ending is a full stop, such as ‘Biol.’. These words are added into the list of honorifics, but another problem arises. For example, ‘... derived from the D4Cole1e gene. We have identified the human homologue ...’ is not split into two sentences due to ‘gene.’ is seen as an honorific. In order to solve such kind of problem, we use heuristic rules as follows:

Step 1. Searching for a token ending with a full stop. (e.g. ‘gene.’)

Step 2. The initial character of the following token is upper case. (e.g. ‘We’)

Step 3. The token is not placed in a pair of parentheses.

If a certain token agrees with these three rules, the sentence will be split into two sentences.

We use *Penn Treebank Tokenizer*, developed by Robert MacIntyre at University of Pennsylvania in 1995, because the POS tags of our training corpus GENIA 3.02p is based on Penn Treebank. This tokenizer written in sed script will produce Penn

Treebank tokenization on arbitrary raw text, and then we can obtain the tokens whose delimiter is one space character (<http://www.cis.upenn.edu/~treebank/tokenization.html>).

### 3.2.2. Part-of-Speech Tagger

A general-purpose Part-of-Speech Tagger is needed to be trained by biomedical corpus. Our training corpus is GENIA 3.02p which contains 65 different types of POS tags.

Our tagger is based on HMM, and three kinds of reasoning methods, which are forward procedure, backward procedure and Viterbi algorithm. Given a token sequence  $T_1^n = t_1 t_2 \dots t_n$ , the goal is to find the optimal POS tag sequence  $P_1^n = p_1 p_2 \dots p_n$  such that

$$\arg \max_P \log Pr(P_1^n | T_1^n) = \sum_{i=1}^n (\log Pr(t_i | p_i) + \log Pr(p_i | p_{i-1})) \quad (3-1)$$

Where  $n$  is length of a sentence,  $Pr$  is a probability function, and  $Pr(P_1^n | T_1^n)$  is the probability of a tag sequence  $P_1^n$  that corresponds to a token sequence  $T_1^n$ . Table 3-2 shows the result. In order to train a better model to tag the articles other than GENIA 3.02p, we use the all the corpus as training set. It is found that the accuracy is better than 10-fold cross validation about 2%. Therefore, the POS tagger is constructed using forward procedure and the model is trained by all data.

10-fold cross validation						
Reasoning	Forward		Backward		Viterbi	
Total #	Corr. #	Accuracy	Corr. #	Accuracy	Corr. #	Accuracy
49534	46996	94.88%	46812	94.50%	46776	94.43%
50077	47555	94.96%	47326	94.51%	47306	94.47%
47679	45199	94.80%	45038	94.46%	44967	94.31%
49299	46769	94.87%	46584	94.49%	46572	94.47%
48818	46323	94.89%	46104	94.44%	46104	94.44%
48781	46305	94.92%	46154	94.61%	46117	94.54%
49578	47042	94.88%	46876	94.55%	46848	94.49%
49495	47003	94.97%	46844	94.64%	46753	94.46%
47722	45141	94.59%	45044	94.39%	44985	94.26%
49486	46843	94.66%	46679	94.33%	46577	94.12%
Average		94.84%		94.49%		94.40%

Table 3-2: The result of 10-fold cross validation in GENIA 3.02p.

### 3.3. Term Variants Resolution

A *variant* is defined as text occurrence that is conceptually related to original terms [Jacquemin C. and Tzoukermann E., '97]. In English, term variants can be separated into three classes:

- (1) **Coordinations:** For example, '*endolymphatic duct and sac*' is a coordination variant of '*endolymphatic sac*'. In GENIA 3.02p, there are 1,595 coordination variants, and 8.04% sentences contain variants.
- (2) **Substitutions:** For example, '*inflammatory sinonasal disease*' is a substitution variant of '*inflammatory disease*'.
- (3) **Permutations:** For example, '*addition of calcium*' is a permutation variant of '*calcium addition*'.


We solve coordination variants only, because GENIA corpus provides the information to train our resolution.

Coordination variants in the corpus can be summarized as three types. In order to label the patterns of coordination variants, we define four types of terms shown in

Table 3-3. Taking the coordination variant ‘91 and 84 kDa proteins’ for example. Where ‘91’ and ‘84’ are the ellipsis terms, ‘kDa proteins’ is the tail term, and ‘and’ is the type term. Ellipsis term is the remaining part of the omitted named entity. Head term is the term in front of the omitted named entity. Tail term is the term in back of the omitted named entity. The head term and tail term will be reproduced in expanding stage. Type term is the conjunction in coordination variants.

Symbol	Description	Amount	Often Seen Terms
#	Ellipsis Term	1,676	B; T; immune; nervous
H	Head Term	260	human; STATs; position
T	Tail Term	522	cels; genes; mRNA
R	Type Term	11	and; or; but not

Table 3-3: Notation of coordination variants.



		Regular Expression	Example
Type 1	Original	$H\#(R\#)^+$	human chromosomes 11p15 and 11p13
	Expanded	$(H\#R)^+H\#$	human chromosomes 11p15 and human chromosomes 11p13
Type 2	Original	$\#(R\#)^+T$	c-fos, c-jun, and EGR2 mRNA
	Expanded	$\#T(R\#)^+T$	c-fos mRNA, c-jun mRNA, and EGR2 mRNA
Type 3	Original	$H\#(R\#)^+T$	human T and B lymphocytes
	Expanded	$(H\#TR)^+H\#T$	human T lymphocyte and human B lymphocyte

Table 3-4: Original pattern, expanded pattern, and examples.

Table 3-4 lists three types of original regular expressions and their corresponding expansions, and in which #, H, T, and R indicate ellipsis/head/tail/type terms. GENIA corpus is divided into training set and testing set that occupy 90% and 10% respectively. The statistics of coordination variants on the training and testing data is shown in Table 3-5.

	# of sentences	# of sentences with coordination variants	# of ellipsis patterns	Percentage of sentences with coordination variants
Training Data	16,684	1,329	1,421	7.97%
Testing Data	1,850	165	174	8.92%

Table 3-5: The statistics of coordination variants on training and testing data.

In baseline experiment, all terms are grouped in a cluster. In order to boost the precision, we design a simple cluster algorithm to group the terms. Let  $(H_i, H_j)$  co-occur in coordination variant, and  $(H_i, H_k)$  co-occur in another one. Then we put  $H_i$ ,  $H_j$  and  $H_k$  into one cluster. We do the clustering procedure recursively, and the statistics shown in Table 3-6.

	Cluster #	Term #	Avg. # of Terms per Cluster
Head Term	145	260	1.79
Tail Term	178	522	2.93
Ellipsis Term	260	1,676	6.45

Table 3-6: The statistics of cluster and term number.

To distinguish the relatedness degree of the terms in the same cluster, the distance between any two terms is computed by applying Floyd-Warshall algorithm. If  $(H_i, H_j)$  co-occur in a phrase and  $(H_i, H_k)$  co-occur in another one but  $(H_j, H_k)$  do not co-occur, and then the  $dist(H_j, H_k) = 2$ .

In our term clusters, the maximum distance between two terms is five. So we implement the variants identification and expansion procedures with the distance range between one and five. Table 3-7 shows that we can get the best result when the distance is equal to one, because instances co-occurring in training set can cover most co-occurrence in testing set. If we relax distance threshold, it can resolve more

instances but most of them are false-positive instances.

	dist.	tp + fn	tp + fp	tp	Recall	Precision	F-Score
Baseline	N/A	165	370	152	92.12%	41.08%	56.82%
Term Clustering	unlimited	165	330	151	91.52%	45.76%	61.01%
	5	165	330	151	91.52%	45.76%	61.01%
	4	165	330	151	91.52%	45.76%	61.01%
	3	165	330	151	91.52%	45.76%	61.01%
	2	165	330	151	91.52%	45.76%	61.01%
	1	165	262	147	89.09%	56.11%	68.85%

Table 3-7: Accuracy of coordination variants identification in GENIA 3.02p.

The same strategy is applied to SRC in which distance is set to one and unlimited respectively. Table 3-8 shows that distance set to one is better than unlimited, and most of resolved instances are not protein entities but biomedical ones.

	GENIA	SRC	
	dist. = 1	Unlimited dist.	dist. = 1
Original Protein Entities	32,525	31,977	31,977
Resolved #	2,445	1,268	1,043
Entities after Expansion	32,876	31,992	31,994

Table 3-8: The number of protein names after term variants resolution and the number of resolved instances.

### 3.4. Noisy Filtering

Intuitively, protein entities do not appear in citation, web link, section title and abstract truncated message, so we use heuristic rules to filter noises. To identify citations, the pairs of parentheses and square brackets are seen as candidates. Then a candidate agrees one of the following rules:

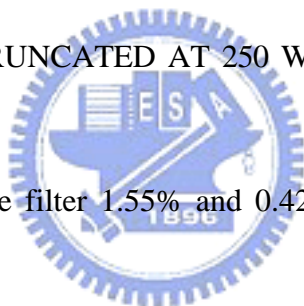
- If a candidate contains ‘et. al’.
- If a candidate contains ‘( n )’, where n is year between 1900 and 2009.
- If a candidate contains author names, such as ‘Krupinski, J.’, ‘J. Krupinski’, ‘Bakalyar, H. A.’ and ‘H. A. Bakalyar’.

We recognize web links if a token ‘http’ is found and the next token is ‘:’. Moreover, the following tokens consist of ‘/’, ‘-’, ‘.’ and alphabet. For instance, three tokens are ‘http’, ‘:’, and ‘//www-genome.wi.mit.edu/’.

Section titles appear at the beginning of a sentence, and tokens of section titles are all capitalized characters. Besides, the ending token of a section title is ‘:’, and an example is ‘FRAGMENT :’.

Some abstracts are truncated in MEDLINE and given a message. The message looks like ‘( ABSTRACT TRUNCATED AT 250 WORDS )’, where ‘250’ may be another number.

After these processes, we filter 1.55% and 0.42% tokens in SRC and GENIA respectively (Table 3-9).



Noisy Types	SRC		GENIA	
	Tagged Token	Percentage	Tagged Token	Percentage
Citation	10,648	1.44%	1,434	0.29%
Web Link	3	0.00%	6	0.00%
Section Title	403	0.05%	454	0.09%
Abstract Truncated Message	413	0.06%	189	0.04%

Table 3-9: The statistic of the number of tokens in the defined regions.

(There are 740,001 and 490,469 tokens in the SRC and GENIA respectively.)

# Chapter 4

## Named Entity Extraction and its Results

We run our named entity extractors on Microsoft Windows 2000 Professional, and database system is Microsoft SQL Server 2000 Personal Edition. We make comparisons with two corpora namely SRC and GENIA 3.02p, and the corpora are divided into training set (90%) and testing set (10%) individually.



### 4.1. Rule-based Named Entity Extraction

Protein entities in training set are composed of core, function and predefined terms. Core terms show the closest resemblance to regular proper names. Function terms describe the functions or characteristics of a protein. Table 4-1 shows predefined terms namely specifier, amino acid and unit.

The frequent regular expressions of protein entities are 'C<sup>+</sup>' and 'F<sup>+</sup>C<sup>+</sup>', where 'C' is core term and 'F' is function term. In SRC and GENIA, the most frequent pattern 'C<sup>+</sup>' occupies 64.90% and 58.36% respectively, and 'F<sup>+</sup>C<sup>+</sup>' occupies 10.99% and 5.15%.



Types	Description	Example
Specifier	Combine numerical characters and a alphabet	1a, 1b, 1c
	Number	1, 2, 3
	Greek letters	alpha, beta, gamma
Amino Acid	The 20 amino acids found in proteins	Gly, Ala, Val
Unit	Units	microM, %, UV

Table 4-1: The three predefined terms.

We define two types of function terms as head function term and tail function term, depending on the position they appear. In our observation, 58.48% head function terms will appear before an initial uppercase token. For example, ‘transcription factor E2F-1’, ‘transcription factor GATA-4’ and ‘transcription factor Sp1’ are three protein entities. Similarly, 74.07% tail function term will appear after an initial uppercase token or a specifier. For example, ‘ACC deaminase’, ‘ACC oxidase’, ‘ACC synthase’ and ‘colicin A immunity protein’, where the shaded terms are function terms. Then we obtain 217 distinct head function terms and 127 distinct tail function terms.

All not explicitly classified are defined as core terms in which continuous English letters are seen as common strings, and these strings are useful for identifying unknown words. For example, a common string ‘CD’ is acquired from a core term ‘CD23’, and then an unknown word ‘CD25’ will be seen as a core term because of its common string is also ‘CD’. In our token list, 3,422 core terms can recover 627 unknown tokens.

Extraction is done by six steps, and first three steps use predefined terms, core terms, and function terms to produce the candidates. If a token is one of the terms, it will be annotated. Adjacent annotated tokens will be seen as a candidate or a chunk, and they will be confirmed or trimmed by step 1 to 3.

### Step 1: boundary confirmation

It is impossible for some POS's to appear on the boundary. We scan the chunk forward (left to right) and backward (right to left) to fix the boundary. The usage of parentheses must be pair-wise, so the irregular parentheses will be removed. For example, a chunk '1 ) IL-2' has to split into '1' and 'IL-2'. The procedure described as follow:

- remove unmatched parentheses.
- the chunk's first POS tag can be one of the set { '(', 'CD', 'JJ', 'NN', 'NNS', 'VBN' }.
- the chunk's last POS tag can be one of the set { ')', 'CD', 'JJ', 'NN', 'NNS', 'VBN' }.
- remove the pair of parentheses when a chunk enclosed.



### Step 2: remove invalid single-token chunks

The following conditions are used to check whether single-token chunks are valid or not. If one of the conditions is matched, the chunk will be regarded as invalid and be removed:

- The characters of a token are all lower case, and the token is not a protein entity in training data. For example, a token 'major' is meaningless.
- It is a predefined term. For example, a token '12' cannot represent an entity.

### Step 3: remove invalid multi-token chunks

To remove invalid multi-token chunks, it needs more evidence. We propose domain independent rules to filter the chunks. A chunk will be removed if it composes of the followings:

- The predefined terms, such as ‘1’, ‘2’ and ‘3’
- The single uppercase English letters, such as ‘A’, ‘B’ and ‘C’.
- The punctuation marks, such as ‘,’ , ‘(’ and ‘)’
- The conjunctions, such as ‘and’ and ‘or’

After the three steps, we remove 68.21% and 52.63% invalid tokens in SRC and GENIA respectively (Table 4-2).

Corpus	Invalid #	Remove #			Filter Rate
		Total	Corr.	Corr. Rate	
SRC	13,451	9,175	9,045	98.58%	68.21%
GENIA	8,846	4,656	4,513	96.93%	52.63%

Table 4-2: The effect of term-based stage (step 1 to 3).

The later three steps aim to acquire precise protein entities as many as possible, so three pattern rules are proposed. Step 4 is to mine the tokens in the preceding and following positions of a protein entity. Fifthly, we want to filter some candidates to boost precision. The sixth one employs syntactic rules to discover some protein names. The rules are generated by applying statistical information yielded from training set.

Step 4: mine the tokens surrounding protein entities

The pattern is formulated as ‘<T<sub>-2</sub>, T<sub>-1</sub>, #, T<sub>1</sub>, T<sub>2</sub>>’, where ‘#’ is token’s number of the protein entity, and the token ‘T<sub>i</sub>’ is the *i*th token relative to the protein entity. Two measurements namely, *confidence* and *occurrence* are used to justify the usefulness of the patterns. *Confidence* means the ratio of the number of correct instances divided by the number of all instances in training data, and *occurrence* is the number of all instances in training data. Patterns are selected whenever their *occurrence* and *confidence* are greater than one and 0.8 respectively, because our system is expected to achieve 80% correct rate, which is the ratio of the number of

correct instances divided by the number of all retrieved instances.

T <sub>-2</sub>	T <sub>-1</sub>	#	T <sub>1</sub>	T <sub>2</sub>	Confidence	Corr. Inst.	Occurrence
receptor	(	1	)	.	0.95	20	21
protein	(	1	)	,	0.94	16	17
factor	(	1	)	,	0.90	18	20
protein	(	1	)	.	0.89	8	9

Table 4-3: The examples of the patterns ' $\langle T_{-2}, T_{-1}, \#, T_1, T_2 \rangle$ '.

#### Step 5: mine the bag-of-word surrounding protein entities

We collect preceding two token and following two token surrounding a protein entity. The *non-confidence* is used to filter the candidates and it is the number of negative instances divided by the number of all instances. Patterns are recognized whenever non-confidence greater than 0.8, because our system is expected to yield 80% correct rate. Table 4-4 gives some examples. One should notice that the candidate with higher *non-confidence* should be removed.

4 bag-of-word				Non-Conf.	Neg. Inst.	Occurrence
	of	the	the	0.91	10	11
in	of	region	the	0.89	8	9
,	,	cells	in	0.80	4	5
.	site	the	to	0.8	4	5

Table 4-4: The examples of the 4 bag-of-word collected from the surroundings of protein entities.

#### Step 6: employ syntactic rules

Hypernym may appear in front of hyponyms [Hearst, '92], and the most common pattern is ' $NP_0$  such as  $\{NP_1, NP_2, \dots, (and|or)\} NP_n$ '. We aim at 'such as' and 'e.g.' and their preceding tokens which provide important clues: '... proteins, such as CBL

and VAV, were phosphorylated on ...', for example. First, we search for 'such as', and then the preceding token 'proteins' tells us the following is a list of protein names. Therefore, we can identify the protein names 'CBL' and 'VAV'. In addition to the clue token 'proteins', we train a list of preceding tokens while these clue tokens appear in training set (Table 4-5).

activated factors	kinases	proteases
activation	lymphocytes	protein
cytokines	lymphokines	proteins
effectors	mediators	receptor
enzymes	molecule	receptors
eosinophils	molecules	stimulus
isoforms	oncoproteins	transcription factors

Table 4-5: The list of significant clue tokens.

The model performance is evaluated in terms of *precision(P)*, *recall(R)* and *F-score(F)* which is  $2PR/(R+P)$ . To present performance of rule-based systems, we use the notations of correct matching defined in [Olsson et al., '02]:

**Sloppy:** Any proposed token matches some tokens of the answer key. For example,

'CD28' vs. 'CD28 surface receptor'.

**Protein Name Parts (PNP):** Each proposed token matches any token of the answer key. For example, 'activation of the CD28 surface receptor' vs. 'CD surface receptor'.

**Strict:** The proposed hit matches one answer key exactly. For example, 'IL-2' vs. 'IL-2'.

**Boundary:**

**Left:** The leftmost proposed token matches a left boundary in the answer key. For example, 'CD28' vs. 'CD28 surface receptor'.

**Right:** The rightmost proposed token matches a right boundary in the answer key. For example, ‘activation of the CD28 surface receptor’ vs. ‘CD28 surface receptor’.

**Left or Right (LorR):** One of the boundaries matches the one of the answer key. This notation is the union of **Left** and **Right**.

Table 4-6 shows that the strict measure can yield 51%-52% F-Score. It also shows that the terms, coming from SRC, are adaptable, because the performance in SRC and GENIA are almost the same. Table 4-7 shows the improvement is obvious after steps 1 to 3, but steps 4 to 6 have a little effect. On the other hand, the precision can be boosted obviously but not much for recall.

	Notation	tp + fn	tp + fp	tp	Recall	Precision	F-Score
	SRC	SLOPPY	3,234	4,782	2,987	92.36%	62.46%
PNP		3,234	4,782	2,859	88.40%	59.79%	71.33%
STRICT		3,234	4,782	2,077	64.22%	43.43%	51.82%
LEFT		3,234	4,782	2,620	81.01%	54.79%	65.37%
RIGHT		3,234	4,782	2,363	73.07%	49.41%	58.96%
LorR		3,234	4,782	2,907	89.89%	60.79%	72.53%
	Notation	tp + fn	tp + fp	tp	Recall	Precision	F-Score
	GENIA	SLOPPY	3,451	4,923	3,010	87.22%	61.14%
PNP		3,451	4,923	2,837	82.21%	57.63%	67.76%
STRICT		3,451	4,923	2,123	61.52%	43.12%	50.70%
LEFT		3,451	4,923	2,765	80.12%	56.16%	66.04%
RIGHT		3,451	4,923	2,296	66.53%	46.64%	54.84%
LorR		3,451	4,923	2,938	85.13%	59.68%	70.17%

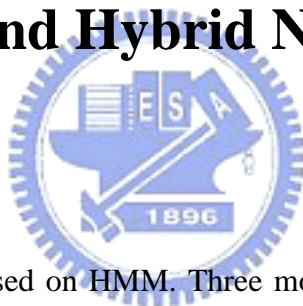
Table 4-6: The rule-based result in SRC and GENIA.

SRC	Procedure	tp + fn	tp + fp	tp	Recall	Precision	F-Score
	Step 1	3,234	10,480	2,051	63.42%	19.57%	29.91%
	Step 2	3,234	5,493	2,043	63.17%	37.19%	46.82%
	Step 3	3,234	4,911	2,040	63.08%	41.54%	50.09%
	Step 4	3,234	4,977	2,104	65.06%	42.27%	51.25%
	Step 5	3,234	4,781	2,077	64.22%	43.44%	51.83%
	Step 6	3,234	4,782	2,077	64.22%	43.43%	51.82%
GENIA	Procedure	tp + fn	tp + fp	tp	Recall	Precision	F-Score
	Step 1	3,451	7,911	2,160	62.59%	27.30%	38.02%
	Step 2	3,451	5,173	2,129	61.69%	41.16%	49.37%
	Step 3	3,451	5,082	2,127	61.63%	41.85%	49.85%
	Step 4	3,451	5,164	2,155	62.45%	41.73%	50.03%
	Step 5	3,451	4,915	2,120	61.43%	43.13%	50.68%
	Step 6	3,451	4,923	2,123	61.52%	43.12%	50.70%

Table 4-7: The intermediate results of rule-based approach.

## 4.2. Statistical and Hybrid Named Entity

### Extraction



The statistical approach is based on HMM. Three models, traditional model, mutual information model and concise model, are examined and a back-off model is also presented. In hybrid extraction system, we put the result of rule-based named entity extraction, and this feature will boost about 1% F-score.

#### 4.2.1. Features Extraction

Internal features indicate those surface clues in tokens (e.g. initial character is upper case), external features indicate the external information associated with tokens (e.g. POS tags), and global features are significant information from training set (e.g. 'protein' indicates that a chunk is a protein entity).

Internal features associate with tokens' characteristics or surface, such as initial upper case and all upper case. These features are shown in Table 4-8, and we use the conjunction of these features. For example, features INIT\_UPPER, SUFFIX\_NUM, LETTER\_DIGITAL, and CONTAIN\_HYPHEN will be assigned to 'BK-2'. Moreover, we consider features not only current token but also preceding token in HMM.

NO	Feature Name	Description	Example
1	INIT_UPPER	The initial character is upper case.	BK-2
2	INIT_LOWER	The initial character is lower case.	c-551
3	INIT_NUM	The initial character is number.	5-HT1B
4	INIT_SYMBOL	The initial character is symbol.	-p1
5	SUFFIX_NUM	The suffix is number.	MDBP-2-H1
6	CONTAIN_GREEK	The token contains Greek letter.	3beta-hydroxysteroid
7	LETTER_DIGITAL	There are letters before number.	A43
8	TWO_CAPS	There are more than two capitalization.	RasHua
9	ALL_UPPER	All characters are upper case.	ALP
10	ALL_LOWER	All characters are lower case.	bombesin
11	NUM	The token is a number.	35 kDa protein
12	OTHER_SINGLE_SYMBOL	It is a symbol, but not "- [ ] : ; % ( ) , . "'	
13	CONTAIN_HYPHEN	The token contains hyphen.	5-HT1B
14	SINGLE_UPPER	The token is a single upper character.	<u>A</u> protein
15	CONTAIN_SLASH	The token contains slash.	C/EBP

Table 4-8: The internal features, and their descriptions and examples.

Besides, we also consider the prefix and suffix string, because they benefit the performance in our studies. We take the most frequent 1,000 three-character prefixes and suffixes strings, and Table 4-9 shows the top 20.

Internal Features	Example
Prefix	pro, tha, gen, seq, wit, con, fro, res, ami, aci, com, str, pre, the, sub, act, thi, exp, alp, tra
Suffix	ion, ing, ase, ted, ein, hat, nce, ith, ent, rom, ine, ate, ity, ene, ide, nal, ins, ons, ino, ain

Table 4-9: The top 20 examples of prefix and suffix strings.



External features are those features extracted not from the components of entities, such as POS tags and BIO tags of rule-based approach. Our classifier can locate protein entities according to POS tags, because tokens of protein entities are normally tagged as nouns. Similarly, the output of rule-based approach is associated with protein entities.

Token	Functions	of	cyclin	A1	in	the	cell	cycle	and	its	interactions
POS	NNS	IN	NN	NN	IN	DT	NN	NN	CC	PRP\$	NNS
R_BIO	O	O	B	I	O	O	O	O	O	O	O
Token	with	transcription	factor	E2F-1	and	the	Rb	family	of	proteins	.
POS	IN	NN	NN	NN	CC	DT	NN	NN	IN	NNS	.
R_BIO	O	B	I	I	O	O	O	O	O	O	O

Table 4-10: A sentence and its corresponding external features. (Where 'R\_BIO' is the result of rule-based named entity extraction.)



Global features are the features extracted from whole training corpus by using statistical method such as Chi-square. Chi-square test is a skill for hypothesis testing of difference. The essence of the test is to compare the observed frequencies with the expected frequencies for independence. The features are usually the significant terms with discrimination to identify the target entities.

To select significant nouns, we use chi-square to measure a token. The simple form of 2-by-2 chi-square test show as following:

$$c^2 = \frac{N(O_{11}O_{22} - O_{12}O_{21})^2}{(O_{11} + O_{12})(O_{11} + O_{21})(O_{12} + O_{22})(O_{21} + O_{22})}$$

Where  $N$  is the number of tokens in the corpus,  $O_{11}$  is the number of specific token in protein name,  $O_{12}$  is the number of other tokens in protein name,  $O_{21}$  is the number of specific token not in protein name, and  $O_{22}$  is the number of other tokens not in protein name.

We run complete-link clustering algorithm to group top 500 nouns, and window size is three sentences. Then we reduce dimensions to 214 and 142 clusters in SRC and GENIA respectively.

Global Features	Example
Nouns	factor, NF-kappa, B, protein, receptor, NF-kappaB, IL-2, alpha, factors, transcription, proteins, kinase, receptors, AP-1, kappa, IL-4, I, cytokines, TNF-alpha, domain

Table 4-11: The significant nouns according to chi-square estimation.

## 4.2.2. HMM Modeling

Given a token sequence  $T_1^n = t_1 t_2 \dots t_n$ , the goal is to find an optimal state sequence  $S_1^n = s_1 s_2 \dots s_n$  that maximizes  $\log Pr(S_1^n | T_1^n)$  which is the logarithm probability of state sequence  $T_1^n$  corresponding to the given token sequence  $S_1^n$ .

A) Traditional HMM:

We apply Bayes's rule:

$$Pr(S_1^n | T_1^n) = \frac{Pr(S_1^n, T_1^n)}{Pr(T_1^n)} \quad (4-1)$$

and then we have

$$\arg \max_s \log Pr(S_1^n | T_1^n) = \arg \max_s (\log Pr(T_1^n | S_1^n) + \log Pr(S_1^n)) \quad (4-2)$$

We assume conditional probability independence and consider preceding state:

$$Pr(T_1^n | S_1^n) = \prod_{i=1}^n Pr(t_i | s_i) \quad (4-3)$$

$$Pr(S_1^n) = \prod_{i=1}^n Pr(s_i | s_{i-1}) \quad (4-4)$$

and the equation (4-2) can be rewritten:

$$\arg \max_s \log Pr(S_1^n | T_1^n) = \arg \max_s \left( \sum_{i=1}^n (\log Pr(t_i | s_i) + \log Pr(s_i | s_{i-1})) \right) \quad (4-5)$$

### B) Mutual Information HMM:

A mutual information HMM was presented in [Zhou and Su, '02] where F-score are 96.94% and 94.28% in MUC-6 and MUC-7 respectively. Different from traditional HMM, the goal is to maximize the equation:

$$\arg \max_s \log Pr(S_1^n | T_1^n) = \arg \max_s \left( \log Pr(S_1^n) + \log \frac{Pr(S_1^n, T_1^n)}{Pr(S_1^n) \cdot Pr(T_1^n)} \right) \quad (4-6)$$

In order to simplify the computation, mutual information independence is assumed:

$$MI(S_1^n, T_1^n) = \sum_{i=1}^n MI(s_i, T_1^n) \quad (4-7)$$

or

$$\log \frac{Pr(S_1^n, T_1^n)}{Pr(S_1^n) \cdot Pr(T_1^n)} = \sum_{i=1}^n \log \frac{Pr(s_i, T_1^n)}{Pr(s_i) \cdot Pr(T_1^n)} \quad (4-8)$$

Applying it to equation (4-6), we have:

$$\arg \max_s \log Pr(S_1^n | T_1^n) = \arg \max_s \left( \log Pr(S_1^n) - \sum_{i=1}^n \log Pr(s_i) + \sum_{i=1}^n \log Pr(s_i | T_1^n) \right) \quad (4-9)$$

### C) Concise HMM:

The concise HMM is based on the idea of maximize the fundamental term  $\log Pr(S_1^n | T_1^n)$ . In the equation (4-9), the terms  $\log Pr(S_1^n)$  and  $\sum_{i=1}^n \log Pr(s_i)$  carry no significant meaning, because the weak probabilities of states and state transitions are merely 3-by-3 and 3-by-1 matrices respectively. Thus, concise HMM is to maximize the equation:

$$\arg \max_s \log Pr(S_1^n | T_1^n) = \arg \max_s \sum_{i=1}^n \log Pr(s_i | T_1^n) \quad (4-10)$$

The concise HMM is incorporated with a back-off model. This is because the concise HMM does not consider HMM's state transition, and a back-off model is a

relaxed probability model whose precision is in decreasing order. Another issue is state transition probability which is the probability of a state transforming into another one, and we should put previous state in the model to ensure correct state induction.

### 4.2.3. Back-off Modeling

Since our system is based on HMM with many features, it is possible to train a high accuracy probability model. However, it is not enough to cover all data, so the data sparseness problem arises. To overcome this problem, we use a back-off model and it aims at the token sequence  $T_1^n$  in  $Pr(S_1^n | T_1^n)$  or  $Pr(s_i | T_1^n)$ .  $T_1^n$  represents not only a token sequence but also its internal, external and global features. Then we define two back-off levels:

A) First level is based on different combinations of tokens and their features, and  $T_1^n$  will be assigned in the descending order:

1.  $\langle s_{-1}, t_{-1}, t_0, f_0 \rangle$
2.  $\langle s_{-1}, t_0, f_0 \rangle$
3.  $\langle s_{-1}, t_{-1}, f_0 \rangle$
4.  $\langle s_{-1}, f_0 \rangle$

Where ' $f_i$ ' represents internal, external and global features. ' $t_i$ ' is a token, ' $s_i$ ' expresses a HMM state, and ' $i$ ' is the  $i$ th one relative to current token.

B) Second level is based on different combinations of features, and ' $f_i$ ' in first level is assigned in the descending order:

1.  $\langle f_i^I, f_i^E, f_i^G \rangle$
2.  $\langle f_i^I, f_i^E \rangle$
3.  $\langle f_i^I \rangle$

Where  $f_i^I$ ,  $f_i^E$  and  $f_i^G$  represent internal, external and global features respectively.

We implemented traditional, mutual information, and concise ones. Then we use same back-off models within concise and mutual information HMM, but not traditional HMM. Table 4-12 shows that concise HMM with rule-based features (i.e. Concise-Hybrid HMM) can yield the best result. Traditional HMM also obtains good F-score, but the recall is not good enough. The reason is that we choose a severe probability model to get the best F-score. The performance of mutual information HMM is the worst, because the back-off model is to optimize concise HMM.

SRC	HMM	tp + fn	tp + fp	tp	Recall	Precision	F-Score
	Concise	3,234	2,953	2,355	72.82%	79.75%	76.13%
Concise - Hybrid	3,234	2,949	2,391	73.93%	81.08%	77.34%	
MI	3,234	3,439	2,384	73.72%	69.32%	71.45%	
Traditional	3,234	2,396	2,086	64.50%	87.06%	74.10%	
GENIA	HMM	tp + fn	tp + fp	tp	Recall	Precision	F-Score
	Concise	3,451	3,285	2,553	73.98%	77.72%	75.80%
Concise - Hybrid	3,451	3,323	2,596	75.22%	78.12%	76.65%	
MI	3,451	3,415	2,305	66.79%	67.50%	67.14%	
Traditional	3,451	2,863	2,263	65.58%	79.04%	71.68%	

Table 4-12: The comparison between HMM models.

Table 4-13 shows every feature has positive effect ( $f^E > f^I > f^G$ ) in concise HMM, because F-score becomes lower if we subtract any feature. Moreover, concise HMM relies on back-off model because features have slight influence on F-score.

SRC	Features	tp + fn	tp + fp	tp	Recall	Precision	F-Score	Diff.
	All	3234	2949	2391	73.93%	81.08%	77.34%	
	All - $f^G$	3234	2948	2372	73.35%	80.46%	76.74%	-0.60%
	All - $f^E$	3234	2888	2319	71.71%	80.30%	75.76%	-1.58%
	All - $f^I$	3234	2943	2342	72.42%	79.58%	75.83%	-1.51%
GENIA	Features	tp + fn	tp + fp	tp	Recall	Precision	F-Score	
	All	3451	3323	2596	75.22%	78.12%	76.65%	
	All - $f^G$	3451	3304	2576	74.65%	77.97%	76.27%	-0.38%
	All - $f^E$	3451	3231	2483	71.95%	76.85%	74.32%	-2.33%
	All - $f^I$	3451	3250	2511	72.76%	77.26%	74.94%	-1.70%

Table 4-13: The effects of features in concise HMM.

## 4.3. Systems Comparison

### 4.3.1. Experiment with SRC

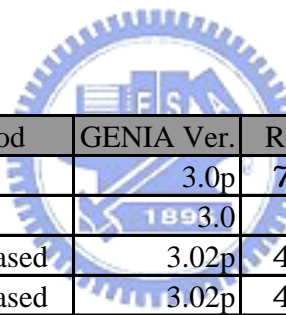
In SRC, we compare our systems with KeX and Yapex. Not only our rule-based approach but also statistical and hybrid ones are better than the two systems. Table 4-14 shows the result, and the unit is F-score. KeX and Yapex yield not bad results in PNP notation because rule-based systems have little ability to identify correct boundaries.

Notation	KeX	Yapex	Our System		
			Rule-based	Statistical	Hybrid
Sloppy	54.85%	60.78%	74.53%	89.12%	89.79%
PNP	48.21%	52.49%	71.33%	80.78%	81.97%
Strict	19.07%	32.37%	51.82%	76.13%	77.34%
Left	32.29%	40.78%	65.37%	85.34%	86.14%
Right	29.53%	47.78%	58.96%	78.36%	79.51%
LorR	42.75%	56.17%	72.53%	88.06%	88.86%

Table 4-14: Comparison between rule-based systems in SRC.

## 4.3.2. Experiment with GENIA Corpus

We run our systems on GENIA version 1.1 and latest version 3.02p published on 19 Aug. 2003. GENIA version 3.02p is based on version 3.0 but errors are fixed. Therefore, we compare with systems, whose GENIA version at least 3.0, developed by Lee et al. [‘03] and Shen et al. [‘03]. Table 4-15 shows that our hybrid approach can yield the best F-score in strict notation. Besides, we can yield a good precision due to incorporation of severe probability models and back-off models. Consider rule-based approaches, KeX and Yapex can yield 72.32% and 65.88% F-score in PNP notation respectively, because it is difficult for rule-based approaches to identify boundaries.



System	Method	GENIA Ver.	Recall	Precision	F-score
(Lee, 2003)	SVM	3.0p	78.80%	61.70%	69.20%
(Shen, 2003)	HMM	3.0			70.81%
KeX	Rule-based	3.02p	43.67%	37.40%	40.29%
Yapex	Rule-based	3.02p	45.06%	50.17%	47.48%
Our System	Rule-based	3.02p	61.52%	43.12%	50.70%
	HMM		73.98%	77.72%	75.80%
	Hybrid		75.22%	78.12%	76.64%

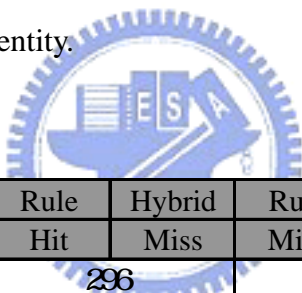
Table 4-15: Comparison with other systems in GENIA version 3.x.

There are 671 abstracts in GENIA version 1.1, and 80 abstracts are selected to be testing set [Kazama et al., ‘02; ‘03]. We compare with Kazama’s systems, and we yield a better result. Compare with the performance in GENIA version 3.02p, and we see one in version 3.02p is better. Because the training set in version 3.02p is larger than that in version 1.1.

System	Method	Recall	Precision	F-score
(Kazama, 2002)	SVM	66.40%	49.20%	56.50%
(Kazama, 2001)	ME	62.10%	49.10%	54.80%
KeX	Rule-based	44.30%	37.39%	40.55%
Yapex	Rule-based	45.39%	51.52%	48.26%
Our System	Rule-based	60.55%	42.83%	50.17%
	HMM	58.12%	63.79%	60.82%
	Hybrid	58.04%	65.19%	61.41%

Table 4-16: Comparison with other systems in GENIA version 1.1.

In our systems, hybrid approach outperforms others, but we still want to know the detail about performance. Table 4-17 shows that hybrid approach can yield about twice entities that rule-based approach cannot identify. Among missing entities, some tagging entities are acceptable. For example, 'human CD80' is a protein entity, but the tag 'CD80' is seen as a wrong entity.



	Rule	Hybrid	Rule	Hybrid	Rule	Hybrid	Rule	Hybrid
	Hit	Hit	Hit	Miss	Miss	Hit	Miss	Miss
SRC	1,781		296		610		547	
GENIA	1,708		415		888		440	

Table 4-17: Comparison with rule-based and hybrid approaches.



# Chapter 5

## Conclusions and Future Works

In this thesis, we propose rule-based and statistical approaches to extract protein entities. Rule-based approach compares with KeX and Yapex, and the result is a slight better than Yapex. However, we use less effort and domain knowledge to establish the model. While it is applied to GENIA corpus, we also obtain a good result.

The proposed global features can boost performance, and the back-off model makes concise HMM to overcome data sparseness problem. Moreover, the feature produced by rule-based approach can be involved to boost a little performance. The result shows that both recall and precision are about 75% in SRC and GENIA 3.02p.

Through term variants resolution, we can identify protein entities hiding behind coordination variants. To enhance the performance of term variants resolution, we use Floyd-Warshall algorithm to compute all-pairs shortest paths that are distance between two terms.

The methods proposed have described in above, and it shows a good result. However, there are several suggestions to improve:

1. Anaphora and Pronoun Resolution:

Both anaphora and pronoun can refer to named entities. To discover all possible relations between entities, we should resolve these phenomena.

2. Coordination Variants Resolution:

In addition to coordination variants, there are many variants such as substitution and permutation variants. If we can detect and resolve these phenomena, a sentence can be understood more easily.

### 3. More Features:

Using back-off models in HMM, data sparseness becomes a little problem, so it is possible to add more features to classify. However, over-fitting problem may arise.

### 4. Over-fitting Problem:

It is difficult to determine a threshold to use appropriate features, so we use features as many as possible. In future, we should pay attention to keep significant features, but nobody knows how many features can achieve the best result.

### 5. Filtering Strategies:

In rule-based approach, we filter the candidates by general heuristic rules. If we have more domain knowledge, it is possible to boost precision.

### 6. Ontology Construction:

After protein entities recognition, it is possible to extract the relations between proteins. Furthermore, pathway and ontology will be constructed by this essential task.



# References

- Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Rapp, B.A., and Wheeler, D.L. (2000). "GenBank." *Nucleic Acids Res* 2000 Jan 1;28(1):15-18.
- Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M.J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S. and Schneider, M. "The Swiss-Prot protein knowledgebase and its supplement TrEMBL in 2003." *Nucleic Acids Res.* 31:365-370(2003).
- Chieu, H.L. and Ng, H.T. (2003). "Named Entity Recognition: A Maximum Entropy Approach Using Global Information." *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, pp 160-163.
- Collier, N., Nobata, C., and Tsujii, J. (2000). "Extracting the Names of Genes and Gene Products with a Hidden Markov Model." *The 18<sup>th</sup> International Conference on Computational Linguistics (COLING 2000)*, pp 201-207.
- Fukuda, K., Tsunoda, T., Tamura, A., and Takagi, T. (1998). "Towards Information Extraction: identifying Protein Names from Biological Papers." *The 3<sup>rd</sup> Pacific Symposium on Biocomputing*, pp 707-718.
- Gaizauskas, R., Demetriou, G., and Humphreys, K. (2000). "Term Recognition and Classification in Biological Science Journal Articles." *Computational Terminology for Medical and Biological Applications Workshop of the 2nd International Conference on Natural Language Processing*, pp 37-44.
- Hanisch, D., Fluck, J., and Mevissen, H. T. (2003). "Playing Biology's Name Game: Identifying Protein Names in Scientific Text." *Pacific Symposium on*

- Biocomputing 2003*, Vol. 8, pp 403-414.
- Hatzivassiloglou, V., Duboue, P. A., and Rzhetsky, A. (2001). "Disambiguating Proteins, Genes, and RNA in Text: A Machine Learning Approach." *Bioinformatics*, Vol. 17, Suppl. 1, pp S97-S106.
- Hearst, M. (1992). "Automatic Acquisition of Hyponyms from Large Text Corpora." *Proceedings of the Fourteenth International Conference on Computational Linguistics, Nantes, France, July 1992*.
- Hou, W.J. and Chen, H.H. (2003). "Enhancing Performance of Protein Name Recognizers using Collocation." *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pp. 25-32.
- Jacquemin C. and Tzoukermann E. (1997). "NLP for term variant extraction: Synergy between morphology, lexicon, and syntax." *In: Strzalkowski T., ed, Natural Language Processing and Information Retrieval. Kluwer, Boston, Mass, 1997*.
- Kazama, J., Makino, T., Ohta, Y., and Tsujii, J. (2001). "A Maximum Entropy Tagger with Unsupervised Hidden Markov Models." *In the Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*. pp. 333-340.
- Kazama, J., Makino, T., Ohta, Y., and Tsujii, J. (2002). "Tuning Support Vector Machines for Biomedical Named Entity Recognition." *Workshop on Natural Language Processing in the Biomedical Domain, Association for Computational Linguistics 2002*, pp 1-8.
- Lee, K.J., Hwang, Y.S., and Rim, H.C. (2003). "Two-Phase Biomedical NE Recognition based on SVMs." *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pp. 33-40.
- Nobata, C., Collier, N. and Tsujii, J. (1999). "Automatic Term Identification and Classification in Biology Texts." *The 5<sup>th</sup> Natural Language Processing Pacific Rim Symposium*, pp 369-374.

- Narayanaswamy, M. and Ravikumar, K. E. (2003). "A Biological Named Entity Recognizer." *Pacific Symposium on Biocomputing 2003*.
- Olsson, F., Eriksson, G., Franzen, K., Asker, L., and Liden P. (2002). "Notions of Correctness when Evaluating Protein Name Taggers." *Proceedings of the 19<sup>th</sup> International Conference on Computational Linguistics*, pp. 765-771.
- Ohta, T., Tateisi, Y., Kim, J.D., Lee, S.Z. and Tsujii, J. (2001). "GENIA corpus: A Semantically Annotated Corpus in Molecular Biology Domain." *In the Proceedings of the ninth International Conference on Intelligent Systems for Molecular Biology (ISMB 2001) poster session*. pp. 68.
- Proux, D., Rechenmann, F., Julliard, L., Pillet, V., and Jacq, B. (1998). "Detecting Gene Symbols and Names in Biological Texts: A First Step toward Pertinent Information Extraction." *Genome Informatics*, pp 72-80.
- Shen, D., Zhang, J., Zhou, G., Su, J., and Tan, C.L. (2003). "Effective Adaptation of a Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain." *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pp. 49-56.
- Takeuchi, K. and Collier, N. (2002). "Use of Support Vector Machines in Extended Named Entity Recognition" *The 19<sup>th</sup> International Conference on Computational Linguistics (COLING 2002)*.
- Takeuchi, K. and Collier, N. (2003). "Bio-Medical Entity Extraction using Support Vector Machines." *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pp. 57-64.
- Torii, M., Kamboj, S., and Vijay-Shanker, K. (2003). "An Investigation of Various Information Sources for Classifying Biological Names." *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pp. 113-120.
- Tsuruoka, Y. and Tsujii, J. (2003). "Boosting Precision and Recall of Dictionary-based

Protein Name Recognition.” *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pp. 41-48.

Wheeler, D.L., Chappey, C., Lash, A.E., Leipe, D.D., Madden, T.L., Schuler, G.D., Tatusova, T.A., and Rapp, B.A. (2000). “Database resources of the National Center for Biotechnology Information.” *Nucleic Acids Res* 2000 Jan 1;28(1):10-4

Wilbur, W. J., Hazard, G. F., Divita, G., Mork, J. G., Aronson, A. R., and Browne, A. (1999). “Analysis of Biomedical Text for Chemical Names: A Comparison of Three Methods.” *The 1999 American Medical Information Association Symposium*, pp 176-180.

Yamamoto, K., Kudo, T., Konagaya, A., and Matsumoto, Y. (2003). “Protein Name Tagging for Biomedical Annotation in Text.” *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pp. 65-72.

Zhou, G. D., and Su, J. (2002). “Named Entity Recognition using an HMM-based Chunk Tagger.” *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL2002)*.

# Appendix A

## An Example of PubMed Reference

[<http://www.ncbi.nlm.nih.gov/>]

```
<?xml version="1.0" ?>
<!DOCTYPE PubmedArticleSet (View Source for full doctype...)>
- <PubmedArticleSet>
  - <PubmedArticle>
    - <MedlineCitation Owner="NLM" Status="Completed">
      <PMID>91354</PMID>
      + <DateCreated>
      + <DateCompleted>
      + <DateRevised>
      - <Article>
        - <Journal>
          <ISSN>0304-8608</ISSN>
          + <JournalIssue PrintYN="Y">
          </Journal>
          <ArticleTitle>Latex fetuin spheres as probes for influenza virus neuraminidase in productively
            and abortively infected cells.</ArticleTitle>
        + <Pagination>
        - <Abstract>
          <AbstractText>Fetuin bound latex spheres do not adhere to the membranes of non-infected
            cells but adhere to those of cells productively infected by fowl plague virus (FPV Dobson
            strain). In contrast, asialo fetuin spheres do not attach to the membranes of
            productively infected cells. Moreover latex fetuin spheres incubated with extracts of
            productively infected cells and extensively washed are specifically enriched in
            neuraminidase activity without any trace of haemagglutinin. These observations
            suggest that viral neuraminidase in the membrane is the site of attachment of the sialic
            acid moieties of fetuin spheres. These neuraminidase sites are detectable when L cells
            are productively infected by a mammalian cell adapted mutant of the Dobson strain
            (FPV-B) but are not detectable on L cells abortively infected by wild type (FPV+).
            However, even in the abortive system, neuraminidase is synthesised de novo as shown
            by its labelling with 14C-glucosamine and by its isolation from labelled extracts of
            infected cells by latex fetuin spheres. These results show that misintegration of viral
            neuraminidase in the plasma membrane of L cells is a feature of abortive infection of
            these cells by the Dobson strain of FPV. However the relationship (if any) of this
            misintegration to abortive infection remains to be established.</AbstractText>
          </Abstract>
        + <AuthorList CompleteYN="Y">
          <Language>eng</Language>
        + <PublicationTypeList>
        </Article>
      + <MedlineJournalInfo>
      + <ChemicalList>
      <CitationSubset>IM</CitationSubset>
      + <MeshHeadingList>
      </MedlineCitation>
    + <PubmedData>
    </PubmedArticle>
  </PubmedArticleSet>
```

# Appendix B

## An Example of Swiss-Prot Entry

[<http://us.expasy.org/sprot/>]

Entry information	
Entry name	IL2_BOVIN
Primary accession number	P05016
Secondary accession numbers	None
Entered in Swiss-Prot in	Release 05, August 1987
Sequence was last modified in	Release 05, August 1987
Annotations were last modified in	Release 41, February 2003
Name and origin of the protein	
Protein name	Interleukin-2 [Precursor]
Synonyms	IL-2 T-cell growth factor TCGF
Gene name	IL2 or IL-2
From	<a href="#">Bos taurus (Bovine)</a> [TaxID: 9913]
Taxonomy	<a href="#">Eukaryota</a> ; <a href="#">Metazoa</a> ; <a href="#">Chordata</a> ; <a href="#">Craniata</a> ; <a href="#">Vertebrata</a> ; <a href="#">Euteleostomi</a> ; <a href="#">Mammalia</a> ; <a href="#">Eutheria</a> ; <a href="#">Cetartiodactyla</a> ; <a href="#">Ruminantia</a> ; <a href="#">Pecora</a> ; <a href="#">Bovidae</a> ; <a href="#">Bovinae</a> ; <a href="#">Bos</a> .
References	
[1]	SEQUENCE FROM NUCLEIC ACID. MEDLINE=86205869; PubMed=3517854; [ <a href="#">NCBI</a> , <a href="#">ExPASy</a> , <a href="#">EBI</a> , <a href="#">Israel</a> , <a href="#">Japan</a> ] <a href="#">Cerretti D.P.</a> , <a href="#">McKereghan K.</a> , <a href="#">Larsen A.</a> , <a href="#">Cantrell M.A.</a> , <a href="#">Anderson D.</a> , <a href="#">Gillis S.</a> , <a href="#">Cosman D.</a> , <a href="#">Baker P.E.</a> ; "Cloning, sequence, and expression of bovine interleukin 2."; <a href="#">Proc. Natl. Acad. Sci. U.S.A. 83:3223-3227(1986).</a>
[2]	SEQUENCE FROM NUCLEIC ACID. MEDLINE=86205870; PubMed=3486415; [ <a href="#">NCBI</a> , <a href="#">ExPASy</a> , <a href="#">EBI</a> , <a href="#">Israel</a> , <a href="#">Japan</a> ] <a href="#">Reeves R.</a> , <a href="#">Spies A.G.</a> , <a href="#">Nissen M.S.</a> , <a href="#">Buck C.D.</a> , <a href="#">Weinberg A.D.</a> , <a href="#">Barr P.J.</a> , <a href="#">Magnuson N.S.</a> , <a href="#">Magnuson J.A.</a> ; "Molecular cloning of a functional bovine interleukin 2 cDNA."; <a href="#">Proc. Natl. Acad. Sci. U.S.A. 83:3228-3232(1986).</a>
[3]	SEQUENCE OF 1-22 FROM NUCLEIC ACID. TISSUE= <a href="#">Thymus</a> ; <a href="#">Anikeeva N.N.</a> , <a href="#">Vinogradova T.V.</a> , <a href="#">Votoshin O.N.</a> ; Submitted (DEC-1989) to the EMBL/GenBank/DDBJ databases.
Comments	
	<ul style="list-style-type: none"><li>• <b>FUNCTION:</b> Produced by T-cells in response to antigenic or mitogenic stimulation, this protein is required for T-cell proliferation and other activities crucial to regulation of the immune response. Can stimulate B cells, monocytes, lymphokine-activated killer cells, natural killer cells, and glioma cells.</li><li>• <b>SUBCELLULAR LOCATION:</b> Secreted.</li><li>• <b>SIMILARITY:</b> Belongs to the IL-2 family.</li></ul>



## Copyright

This Swiss-Prot entry is copyright. It is produced through a collaboration between the Swiss Institute of Bioinformatics and the EMBL outstation - the European Bioinformatics Institute. There are no restrictions on its use by non-profit institutions as long as its content is in no way modified and this statement is not removed. Usage by and for commercial entities requires a license agreement (See <http://www.isb-sib.ch/announce/> or send an email to [license@isb-sib.ch](mailto:license@isb-sib.ch))

## Cross-references

<b>EMBL</b>	M12791; AAA30586.1; -. [ <a href="#">EMBL</a> / <a href="#">GenBank</a> / <a href="#">DDBJ</a> ] [ <a href="#">CoDingSequence</a> ] M13204; AAA21143.1; ALT_INIT. [ <a href="#">EMBL</a> / <a href="#">GenBank</a> / <a href="#">DDBJ</a> ] [ <a href="#">CoDingSequence</a> ] X17201; CAA35062.1; -. [ <a href="#">EMBL</a> / <a href="#">GenBank</a> / <a href="#">DDBJ</a> ] [ <a href="#">CoDingSequence</a> ] X52687; CAA36912.1; -. [ <a href="#">EMBL</a> / <a href="#">GenBank</a> / <a href="#">DDBJ</a> ] [ <a href="#">CoDingSequence</a> ]
<b>PIR</b>	<a href="#">I45913</a> ; I45913.
<b>HSSP</b>	<a href="#">P01585</a> ; 1M49. [ <a href="#">HSSP ENTRY</a> / <a href="#">SWISS-3DIMAGE</a> / <a href="#">PDB</a> ]
<b>InterPro</b>	<a href="#">IPR000779</a> ; Interleukin-2. <a href="#">Graphical view of domain structure.</a>
<b>Pfam</b>	<a href="#">PF00715</a> ; IL2; 1. <a href="#">Pfam graphical view of domain structure.</a>
<b>PRINTS</b>	<a href="#">PR00265</a> ; INTERLEUKIN2.
<b>ProDom</b>	<a href="#">PD003649</a> ; Interleukin-2; 1. <a href="#">[Domain structure / List of seq. sharing at least 1 domain]</a>
<b>SMART</b>	<a href="#">SM00189</a> ; IL2; 1.
<b>PROSITE</b>	<a href="#">PS00424</a> ; INTERLEUKIN_2; 1.
<b>HOVERGEN</b>	<a href="#">[Family / Alignment / Tree]</a>
<b>BLOCKS</b>	<a href="#">P05016</a> .
<b>ProtoNet</b>	<a href="#">P05016</a> .
<b>ProtoMap</b>	<a href="#">P05016</a> .
<b>PRESAGE</b>	<a href="#">P05016</a> .
<b>DIP</b>	<a href="#">P05016</a> .
<b>ModBase</b>	<a href="#">P05016</a> .
<b>SMR</b>	<a href="#">P05016</a> ; 816667DFEA052EDF.
<b>SWISS-2DPAGE</b>	<a href="#">Get region on 2D PAGE.</a>
<b>UniRef</b>	View cluster of proteins with at least <a href="#">50%</a> / <a href="#">90%</a> identity.

## Keywords

**Cytokine; Glycoprotein; Immune response; Signal; Growth factor; T-cell.**

## Features

[Feature table viewer](#)

Key	From	To	Length	Description
<a href="#">SIGNAL</a>	<a href="#">1</a>	<a href="#">20</a>	20	By similarity.
<a href="#">CHAIN</a>	<a href="#">21</a>	<a href="#">155</a>	135	Interleukin-2.
<a href="#">DISULFID</a>	<a href="#">79</a>	<a href="#">127</a>		By similarity.
<a href="#">CARBOHYD</a>	<a href="#">23</a>	<a href="#">23</a>		O-linked (GalNAc...) (By similarity).
<a href="#">CONFLICT</a>	<a href="#">66</a>	<a href="#">66</a>		V -> A (in Ref. <a href="#">2</a> ).

## Sequence information

**Length:** 155 AA [This is the length of the unprocessed precursor]      **Molecular weight:** 17627 Da [This is the MW of the unprocessed precursor]      **CRC64:** 816667DFEA052EDF [This is a checksum on the sequence]

```

      10      20      30      40      50      60
      |      |      |      |      |      |
MYKIQLLSCI ALTLALVANG APTSSSTGNT MKEVKSLLLD LQLLLEKVKN PENLKL SRMH
      70      80      90     100     110     120
      |      |      |      |      |      |
TFDFYVPKVN ATELKHLKCL LEELKLL EEV LNLAPSKNLN PREIKDSMDN IKRIVLELQG
     130     140     150
      |      |      |
SETRFTCEYD DATVNAVEFL NKWITFCQSI YSTMT
```

P05016 in [FASTA format](#)

# Appendix C

## Function terms

acanthin, acetylhydrolase, acid, activating, activator, acyl, adducin, adenylyl, adenylyltransferase, adrenergic, alcohol, aldolase, alkaline, allergen, amidinotransferase, aminopeptidase, amphiphysin, amyloid, anchorin, angiotensin, angiotensinase, anhydrase, annexin, anthopleurin, anthranilate, anticoagulant, antigen, antithrombin, apolipoprotein, apoxin, arginase, arylsulfatase, aspartokinase, aspergillopepsin, attachment, autoregulatory, avicelase, azurin, bdellin, binding, bisphosphatase, blocker, bom, bombolitin, bothropstoxin, botulinum, brain, buforin, calbindin, calcium, calgranulin, capsid, carbonic, carboxylase, carboxypeptidase, cardiac, cardiotoxin, catalase, catalytic, cathepsin, cecropin, chain, channel, chargerin, chondroitinase, chromogranin, chromosomal, chymotrypsinogen, circulin, class, coagulation, coenzyme, cofactor, colicin, collagen, complex, component, concanavalin, congerin, conotoxin, convertase, converting, copine, coproporphyrinogen, curvacin, cyclase, cyclin, cyclohydrolase, cyclophilin, cylicin, cystathionine, cystatin, cysteine, cytochrome, cytolysin, cytosolic, deaminase, decay, defensin, degenerin, dehydratase, dehydrogenase, dendrotoxin, deoxyribonuclease, deoxyribophosphodiesterase, dipeptidyl, dipeptidylpeptidase, discoidin, dismutase, dissociation, ecarpholin, elongation, elongin, endo, endoglucanase, endoglycosidase, endonexin, endonuclease, endopeptidase, endophilin, endoproteinase, endothelial,

enolase, enzyme, epidermal, equinatoxin, esterase, exfoliative, exonuclease, exotoxin, factor, fasciclin, fd, ferredoxin, flavin, flavoprotein, fly, fructokinase, fucosyltransferase, gelatinase, gland, globulin, glucose, glucosidase, glutamate, glutathione, glycine, glycohydrolase, glycophorin, glycoprotein, glycosylase, glyoxalase, golgi, gonococcal, granzyme, group, growth, heavy, helicase, hemoglobin, heparin, hexosaminidase, histone, homolog, hydratase, hydrogenase, hydrolase, hydroxyindole, immunity, immunoglobulin, inducing, inhibitor, isoform, isomerase, isopenicillin, isozyme, kalata, kinase, lamin, laminin, large, leader, lectin, leiuropeptide, leiurotoxin, leukotriene, lig, ligase, ligatoxin, lipid, lipocortin, lipoprotein, lqq, lyase, lysophospholipase, lysozyme, major, makatoxin, malate, malic, mannosidase, mast, matrin, membrane, methylated, microsomal, mitochondrial, molecule, motch, mouse, mt, mutacin, mutase, myelin, myomodulin, myosin, myotoxin, myristoylated, natriuretic, neurexin, neurokinin, neuromedin, neuropeptide, neurotoxin, nicotinamide, nuclear, nuclease, nucleosidase, nucleoside, nucleotidylexotransferase, odorant, oligopeptidase, oncorhyncin, ornithine, orphanin, outer, oxidase, oxidoreductase, paraneoplastic, peak, pectin, pepsin, pepsinogen, peptidase, peptide, permease, peroxidase, phenylethanolamine, phosphatase, phosphatidylserine, phosphodiesterase, phosphoglycerate, phospholipase, phosphorylase, phytochrome, placenta, placental, pol, poly, polymerase, polypeptide, polyprotein, porin, precursor, preferential, preprotachykinin, primase, procarboxypeptidase, procollagen, profilin, proproteinase, prostaglandin, protamine, protease, protein, proteinase, pseudouridine, purine, pyrophosphatase, pyrophosphorylase, receptor, reductase, rennin, replicase, repressor, response, rhodopsin, ribonuclease, ribophorin, ribosomal, ribosyltransferase, ricin, saposin, scylliorhinin, secretogranin, selenoprotein, sema, semaphorin, sensory, serogroup, serum, sialokinin, signal, small, soluble, somatomedin, stefin, sterol, stimulatory,

stoned, strain, streptolysin, stress, structural, substance, substrate, subtilisin, subunit, sulfhydrylase, sulfurylase, superoxide, surface, symporter, synapsin, synaptotagmin, synthase, synthetase, tautomerase, telomeric, tfu, thioether, thiol, thiolase, thiopurine, thioredoxin, thymidylate, tityustoxin, torsin, toxin, transcobalamin, transcription, transferase, transformylase, transporter, troponin, trypsin, tubulin, type, ucn, upstream, uricase, urocortin, uroplakin, urotensin, variant, vascular, venom, vitamin, vitellogenin, von

