Automated Classification and Information Extraction of Biological Text

Jihoon Yang[1], Yoonhee Choi[2], Kiho Hong[3], Junhyung Park[4], and Seonho Kim[5]
Department of Computer Science, Sogang University[1,5]
Digital Media R&D Center, SAMSUNG ELECTRONICS CO.,LTD, Korea[2]
IT Agent Research Lab, LSIS R&D Center, Korea[3], *Diquest Inc., Seocho-gu, Seoul, Korea*[4]
yangjh@sogang.ac.kr[1], *yhyh.choi@samsung.com*[2], *khhong1@lgis.com*[3],
pustar@gmail.com[4], *shkim@lex.yonsei.ac.kr*[5]

Abstract

We present a text mining system for protein-protein interaction extraction. We first label documents as relevant/irrelevant to a given topic with metadata keyword vectors, and then extract protein names based on HMM tagging. Lastly, interactions between protein names are recognized with simple parse rules and a confidence measure. The experimental results show that the proposed method achieves a very promising result, 83.87% F-score for protein-protein interaction extraction from biomedical documents without deep level parsing.

1. Introduction

A huge amount of publications have been pouring out with the rapid growth of biological research. Thus, the efficient access to this increasing information and the newest knowledge discovery through data mining has become important issues in this area.

We suggest a total biological mining system which consists of three components: document retrieval, named entity recognition, and relation extraction between named entities. We first classify documents by using metadata keywords, rather than full text keywords, and then identify protein names appearing in the relevant documents using HMM(Hidden Markov Model) tagging. For the relation extraction, discriminating event verbs are first retrieved through the P-score estimation and then relation pattern triples are extracted on the basis of a confidence measure from the candidate sentences including the event verbs and protein entities.

2. Document classification

For the document classification, we extract keywords from the metadata of a document instead of the entire document. There have been some works on metadata keyword-based document classification.

In this work, we consider the four kinds of metadata forms: 'title', 'author', 'abstract', and 'reference'. We first compare a classification performance according to such meta-textual source combinations by alternating machine learning algorithms. We apply three machine learning algorithms: Naïve Bayes [1], SVM(Support Vector Machine) [7], and TF(Term Frequency)-IDF(Inverse Document Frequency) [8]. In addition, we investigate the classification performance by full text keywords, for a comparison.

Here, we focus on the problem of retrieving documents related to 'apoptosis'. The topic is very important as 'apoptosis' is a process of deliberate life relinquishment by a cell in a multicellular organism. In addition to its importance as a biological phenomenon, its defective apoptotic processes have been implicated in an extensive variety of diseases. We first collected documents through the MEDLINE database and Google search results with the query 'apoptosis'. For the classification learning, domain experts manually classified categories of the documents as 'related' and 'not related'. In this section, we describe the document classification by metadata.

2.1. Preprocessing and metadata extraction

If a document is in XML or HTML format, we can easily extract the metadata from its tag structure. However, because most electronic documents are available in Postscript or PDF formats, it is difficult to directly extract metadata from them. Thus, we convert all files into HTML and unstructured text formats.

† This work was supported by grant No. R01-2004-000-10689-0 from the Basic Research Program of the Korea Science & Engineering Foundation

As mentioned earlier, we use the 'title', 'author', 'abstract', and 'reference' information of a document as its metadata. The title, author, and abstract metadata of a document are used to construct a normal word vector whereas its reference information is for a reference vector.

In general, title information is easily extracted with the font tags of HTML/XML documents. That is, title phrases are usually enclosed by angle brackets for bold or large font tags at the beginning of a document. Since titles of HTML/XML documents are usually written in bold face, those can be easily identified by a regular expression such as '<B> ~ </B>'.

On the other hand, we take into consideration other characteristics besides font tags in the case of author information. In fact, many documents are written by more than one author. Such documents will contain a couple of lines for author descriptions which are represented with several commas and periods. Typically, author names start with capital letters and are joined by commas, and 'and'. Thus, we regard a sentence as an author description if it contains words starting with capital letters, commas or periods occur more than two times. When the author description lines are extracted, each author is separated.

As other metadata, we retrieve abstract and reference information by using explicit word keywords. For the abstract extraction, the keywords such as 'abstract' or 'summary' and their variants are used. 'Reference' and its variants are used for the reference extraction. For example, 'REFERENCES', 'Reference', 'References' can be the lexical variants. First of all, we find where a reference paragraph starts and ends, and then separate each reference by numbers, as references are often listed by numbers.

In this paper, we assume that keywords on metadata can improve classification accuracy.

2.2. Vector representation

We adopt the vector space model and the Naïve Bayes classifier. In the vector space model, documents are represented as vectors of index terms(keywords). However, we use the set of terms as all unique words occurring in the metadata of a document, rather than full text. Thus, a vector for a document is constructed from its metadata. In addition, we consider a reference vector to represent a document as well as the word vector. We assume that the documents with similar references are related. The reference vector is used to evaluate similarities among documents and it is finally integrated into the Naïve Bayes classifier. The similarity between two reference vectors is measured by a cosine similarity.

2.3. Document classification

We first construct different keyword vectors by combining four kinds of metadata. As a result, keyword vectors generated from all metadata: 'title', 'abstract', and 'author' showed a good performance for document classification. Also, the Naïve Bayes [1] outperforms other classifying methods. The Naïve Bayes classifier assigns each document d_i to the category c_j with the highest probability and the probability can be calculated as follows:

$$\arg\max_{c_j \in \mathbf{C}} P(c_j) P(c_j | d_i) \qquad (1)$$

In our work, we categorize documents as two groups: 'relevant' or 'irrelevant' to the topic, 'apaptosis'. We modify equation (1) with the word (keyword) vector, w_i and reference vector, r_i. The term w_{ki} denotes the k^{th} element of a word vector of a document d_i and r_{ki} denotes the k^{th} element of a reference vector of a document d_i. r_{cj} refers to a representative reference vector composed of entire documents which belong to the category c_j.

$$\arg\max_{c_j \in \mathbf{C}} P(c_j) \prod_{k=1}^{|T|} P(w_{ki} | c_j) \prod_{k=1}^{|R|} \cos(r_{ki}, r_{c_j}) \qquad (2)$$

3. Named entity recognition

Like other biomedical terminologies, protein names have many lexical variants such as abbreviations, acronyms, nicknames, official or formal names, or full names. It is very common that protein names have many variants in capitalizations, punctuations, and special characters.

In this manuscript, for SPN[1] extraction, the HMM POS(parts-of-speech) tagging are used. On the contrary, we adopt a probabilistic model for MPN extraction.

3.1. SPN extraction

In order to identify SPNs, we perform POS tagging for a given text with the Brill's tagger [2]. The tagger is trained on the GENIA corpus [9] and the HMM tagging with Viterbi algorithm is performed to extract SPNs. In particular, we consider a further POS tag of GENE for SPNs when tagging documents. For this, we first construct a lexicon from GenBank[2] database words. The words can be tagged as 'GENE'. In general, the naming conventions of proteins are irregular and ambiguous, even though there are some rules for the naming [6]. Furthermore, they contain a lot of special characters such as hyphens, slashes, parentheses, Roman letters, and Greek and Arabic

[1] We call a single word protein name SPN and multiple words protein MPN

[2] http://www.ncbi.nlm.nih.gov/Genbank/index.html

numbers. The names can include ‘and’, ‘or’, or ‘of’, too. Thus, we cannot identify variants of SPNS only with the lexicon.

In order to handle various spelling forms, we attempt a postprocessing after the HMM tagging. We first normalize words by character substitutions. A special character is substituted with a metacharacter ‘#’, a digit with ‘&’, an alphabet with ‘C’ and a Roman letter with ‘?’, respectively. Such normalization allows partial matches to protein names.

3.2. MPN extraction

In general, protein names are composed of several words and an SPN can be expanded to an MPN with neighboring words. However, an MPN may not include an SPN annotated as ‘GENE’ such as ‘tumor necrosis factor’. In order to find out MPNs, we enhance the probability model used in TagGeN [6]. First, if a word is a SEED word, then the word is bidirectionally expanded. For this, we collect 80 SEED words which frequently appear in MPNs. We determine the range of a MPN on the basis of the following probability.

$$P(W_{next} \mid W_{current}, M_{current} = 1) \qquad (3)$$

where W_i represents the features of the word at the position i, W_{next} can be the left and the right words of W_i, and M_i is a binary feature which indicates whether the word at position i belongs to MPN. Initially, the seed words have the value of 1. For a given SEED word, the system bidirectionally expands its neighboring words.

Table 1. Examples of W_i's used in probabilistic models

Left Direction	
Set	*Example*
NN(Noun Class)	Single-chain/**NN** fv/GENE
JJ(Adjective Class)	Human/**JJ** GM-CSF/**GENE** gene/NN
CD(Number Class)	3/**CD** alpha/NN HSD/GENE
GENE(Gene Class)	Huma/JJ GN-CSF/**GENE** gene/NN
…ase	Phospholip**ase**
Roman, Greek Character	Type **II** IL-1R
Word Set (i.e. protein, gene, factor, etc.)	**protein** tyrosine kinase
Right Direction	
Set	*Example*
reporter	beta-globin **reporter**
product	start-1 gene **product**
single character	c-erb **A**
Numerals	IFN-stimulated gene factor 3
…ed	C5a induc**ed** kappa-B
…like	Proximal c-jun TRE-**like** promoter element
…ing	IRF-1 GAS-bind**ing** complex

Some feature examples of W_i are illustrated in Table 1. We distinguish the features for left expansion and the features for right expansion. In general, the left side words of an MPN have various word classes than the right side words and the right words of an MPN contain digits or adjective affixes such as ‘-ed’, ‘-like’, or ‘-ing’.

With the conditional probability of equation (3), we can compute the transition probability from W_i to the next word. We elaborate the method used in TagGeN [6] for the bidirectional expansion of seed words. For the probabilistic model for MPNs, we randomly selected 600 documents from GENIA corpus and MPNs of the documents are manually tagged by domain experts.

4. Interaction extraction

In this paper, an interaction(relation) between identified proteins is represented with a triple like ‘*Protein*(*A*)-*Type*(*interaction*)-*Protein*(*B*)’ and the relation pattern is extracted by discriminating verbs and a confidence value. We first extract the discriminating verbs with the P-score and then extract the associated protein-protein interactions by a confidence score.

Our relation extraction method is based on the co-occurrence approach. In the simple co-occurrence based work, two entities are assumed to have a relationship if they are only mentioned together without being necessarily related in a specific way. That is, the relationship implies that two entities repeatedly occur together or by the presence of some linguistic expressions. However, relations between entities are less predictable by pure co-occurrences of terms in sentences. Thus, we use a specificity measure with a confidence score to ensure that the extracted relations are not too general.

4.1. Discriminating verb extraction

A discriminating verb is extracted as follows:

1. Pre-processing
   We analyze documents by POS tagging and stemming. The verb POS of Brill’s tagger tags verbs are VB(base form), VBN(past participle), and VBZ(3rd person singular present). Suffixes are removed using the Porter stemmer.
2. P- Score estimation
   We design a Bayesian probabilistic model to estimate a P-score of each verb, and then determine the set of discriminating verbs based on the P-scores. This method was proposed to extract a set of words for document classification by Marcotte *et al.* [4]. We apply this method for extracting a set of discriminating verbs. The P-score exhibits how well a verb describes an interaction between proteins. To

compute the P-score, training interaction patterns are required and the score is calculated as follows:

$$P(n \mid N, f) \approx e^{-Nf} \frac{(Nf)^n}{n!} \qquad (4)$$

where n means how many times a verb is used as a protein interaction, N is the total number of words in a document, and f is the total occurrences of each verb. The Poisson distribution can be an alternative for the probability $P(n \mid N, f)$ if N is big enough and f is fairly small.

3. Discriminating verb selection
   We calculate the P-score with respect to every verb, and then choose a set of arbitrary number of words with the highest P-scores. We fix the set as 80 words. For example, 'inhibit', or 'indicate' are retrieved as discriminating verbs. Table 2 shows examples of the discriminating verbs. For a computational convenience, the logarithm of (4) is used. That is, a verb with a low P-score is selected as a discriminating word.

Table 2. Discriminating word example

Number	Word	P-Score
1	inhibit	-2064531
2	indicate	-1906823
3	regulate	-1849474
4	require	-1691767
5	result	-1462374
6	report	-946242
7	detect	-88894

4.2. Protein mutual effect extraction

Based on the discriminating verbs, we extract interactions between proteins. For this, we first retrieve candidate sentences which contain at least two protein names and one discriminating verb. However, due to ambiguities of natural language, it is difficult to recognize the internal structure of a sentence well. Thus, we use simple pattern matching rules to avoid the ambiguity of natural language structures. The steps of extracting protein-protein interaction are as follows:

1. Complex sentence processing
   To avoid the difficulties of parsing of sentences, we used Ono's pattern matching rules. Ono *et al.* [9] and Stapley *et al.* [3] extracted interaction patterns based on simple syntactic rules. The sentences are parsed using the part-of-speech rules in Table 3 that convert a complex sentence into a simple sentence and handle negative sentences.
2. Interaction Extraction
   If there is a pattern like "$Protein_A$-TYPE (discriminating verb)-$Protein_B$" in a sentence, we calculate *Confidence* of the pattern and then add the sentence into the *event* (protein, interaction) set.

Table 3. Interaction pattern extraction rules

<table>
<tr><th>regular expression</th><th>Pattern</th></tr>
<tr><td>$Protein_A$[(,<i>CCDT</i>|(,<i>IN</i>)| : | ;]$Protein_B$</td><td>$Protein_A$, $Protein_B$</td></tr>
<tr><td>$Protein_A$ VB1 $Protein_B$ VB2 CC $Protein_C$</td><td>$Protein_A$ VB $Protein_B$
$Protein_A$ VB $Protein_C$</td></tr>
<tr><td>$Protein_A$* not VB1 * $Protein_B$</td><td>$Protein_A$ not VB $\mathrm{Protein}_B$</td></tr>
<tr><td>$Protein_A$ * but not $Protein_B$</td><td>$Protein_A$not VB $Protein_B$</td></tr>
</table>

The *Confidence* is calculated as follows:

$$Confidende = s(p) + \frac{1}{sd} \qquad (5)$$

where *s(p)* is a binary function which represents whether a pattern *p* is included in a sentence or not, and *sd* is the sum of distances between each protein and a discriminating verb in the sentence. The distance is a number of words which are from a verb to a protein. In the case of 'IL-10 inhibits IFN-gamma-induced ICAM-1 expression in monocytes.', '*IL*-10' and '*inhibit*' have a distance of 2 and '*inhibit*' and '*IFNgamma-induced ICAM*-1 *expression*' have a distance of 1.

5. Experiments

5.1. Document classification

We collected 470 documents with the query 'apoptosis' from the MEDLINE database and Google search results. Among them, 223 papers were classified as relevant documents and 247 papers were classified as irrelevant documents by human experts. For metadata extraction, we converted the documents into two types of formats - HTML and unstructured text. We finally used 458 well-converted documents in our experiment. The performance of the document classification was evaluated by *accuracy*, *precision*, *recall*, and *F-measure* based on a confusion matrix.

We investigated the classification performances with respect to the following metadata combinations: 'title', 'title, author', 'title, author, abstract', and 'abstract'. We applied three machine learning algorithms: Naïve Bayes, SVM(Support Vector Machine), and TFIDF(Term Frequency Inverse Document Frequency). For this, a 10-fold cross validation and the RAINBOW[3] Toolkit of McCallum were used. RAINBOW is a program that supports statistical text classification. For a comparison, the classification using full text keywords was carried out.

As shown in Table 4, the Naïve Bayes classifiers based on keywords, 'title, author, abstract' showed the best classification performance. In the TFIDF

[3] http://www.cs.cmu.edu/mccallum/bow/rainbow/

Table 4. Comparison of performance using full text and metadata

			Naïve Bayes	TFIDF	SVM
metadata	Title	Accuracy	70.4	67.8	64.9
		Precision	70.2	60.3	65.9
		Recall	73.3	90.5	63.1
		F-measure	71.7	72.4	64.5
	Title, Author	Accuracy	75.6	73.8	63.3
		Precision	73.1	69.5	61.5
		Recall	82.6	82.7	76.2
		F-measure	77.6	75.5	68.1
	Title, Author, Abstract	Accuracy	86.7	80.4	70.4
		Precision	90.5	71.2	69.3
		Recall	82.6	85.5	81.8
		F-measure	86.4	77.7	75.0
	Abstract	Accuracy	84.8	79.4	71.7
		Precision	83.3	83.3	70.8
		Recall	87.0	90.9	73.9
		F-measure	85.1	87.0	72.3
All words		Accuracy	84.3	76.2	70.2
		Precision	87.4	92.3	68.1
		Recall	81.8	57.1	76.2
		F-measure	84.5	70.6	71.9

Table 5. Results of classification using reference

	Percentage
Accuracy	74.7
Precision	71.9
Recall	78.2
F-measure	74.9

Table 6. Comparison result

	Accuracy	Precision	Recall	F-measure
title, author, abstract	86.7	90.5	82.6	86.4
similarity of reference	74.7	71.9	78.2	74.9
combining classification	89.4	87.9	90.9	89.4

classification, 'abstract' metadata showed the highest *F-measure*. In conclusion, there is no significant difference among the learning algorithms, but the vector representation considering all metadata, 'title', 'author', and 'abstract' performed well. Furthermore, the vector representation using metadata outperformed the full text keyword vector representation.

5.1.1. Classification using reference

We constructed a representative reference vectors according to each category. As a result, more than 10,000 reference vectors were collected for each category. In this paper, the reference collections were separated into two categories, 'relevant' and 'irrelevant' and then a representative reference vector was constructed with respect to each category. For a given document, two representative reference vectors were compared with its reference vector. We choose the category with the higher similarity value. The result is not good as much as metadata representation, as shown in Table 5.

5.1.2. Combined classification

So far, we obtained the best classification performance with Naïve Bayes over the metadata vectors of 'title, author, abstract'. We combined the method with cosine similarities between reference vectors like equation (2).

Table 6 shows the result of the combined method. As a result, the combined method improved the classification performance about 3% (F-measure) compared with the Naïve Bayes with 'title, author, abstract'.

5.2. Protein name recognition

The test data used for the experiments consisted of 600 papers from the GENIA Corpus. We compared our method with ABGene [6] and TagGeN [5] which are both based on Brill's tagger.

5.2.1. SPN extraction

Table 7 exhibits performances of SPN extraction as changing data size from 100 to 600 by 100 documents. Due to the substring matching method, our system showed a lower accuracy than ABGene, while it achieved a higher F-measure. In addition, our system was significantly faster than others due to protein name hashing and a simplified tagging process.

5.2.2. MPN extraction

In the case of MPN extraction, we evaluated the results, separating exact match and partial match. As shown in Table 8, our approach outperformed TagGeN in MPN extraction.

5.3. Protein interaction extraction

We used 80 discriminating verbs with a high P-score. For this experiment, we selected 100 test sentences including 14 negative, 8 complex sentences. From the sentences, we obtained 139 protein interactions and 83.87% of F-measure as presented in Table 9.

6. Conclusion

In this study, we suggested a total text mining system for protein-protein interaction extraction. We constructed keyword vectors by using the metadata of each document, and classified documents with the Naïve Bayes classifier. In addition, we combined the classifier with the reference similarity measure. For the SPN extraction, the HMM POS tagging were used and

Table 7. Performance of SPN extraction

Dataset System	100	200	300	400	500	600	Average
Accuracy							
Our system	83.28	85.17	84.97	85.10	85.58	85.88	85.00(%)
ABGene	87.40	87.12	87.13	87.19	86.12	87.10	87.01(%)
TagGeN	80.17	82.24	83.51	84.09	84.50	84.91	83.24(%)
F-measure							
Our system	88.78	90.26	90.32	90.66	90.88	90.80	90.28(%)
ABGene	63.74	69.02	62.86	66.79	70.74	71.83	67.56(%)
TagGeN	74.02	78.72	80.83	80.48	81.56	81.90	79.56(%)
Processing time							
Our system	2.81	3.50	4.23	4.85	5.46	6.23(s)	
ABGene	19.01	39.28	56.12	74.31	94.11	113.00(s)	
TagGeN	5913	11925	18777	24970	30979	36324(s)	

Table 8. Performance of MPN extraction

	Recall (%)	Precision (%)	F-measure (%)
Exact Match			
Our system	84.25	86.65	84.84
TagGeN	80.23	87.81	83.84
Partial Match			
Our system	91.56	91.35	91.84
TagGeN	86.51	91.15	88.77

Table 9. Performance of interaction extraction

Precision (%)	Recall (%)	F-measure (%)
76.58	92.70	83.87

a probabilistic model were used for the MPN extraction. Finally, we extracted protein-protein interactions based on discriminating verbs and confidence values of patterns. Experimental results showed that the proposed method achieved an F-measure of 83.87% for protein-protein interaction extraction. The main contribution of our work is the development of a total data mining system, from document classification to relation extraction.

7. Reference

[1] A.McCallum and K.Nigam, "A Comparison of Event Models for Naive Bayes Text Classification", *In Learning for Text Categorization Workshop, National Conference on Artificial Intelligence*, 1998, pp.41-48.

[2] B.Eric, "Some Advances in Transformation-based Part of Speech Tagging", *In AAAI,* 1994.

[3] B.J.Stapley and G.Benoit, "Biobibliometrics: Information Retrieval and Visualization from Co-Occurrences of Gene Names in Medline Abstracts", *In Proceedings of the PSB 2000,* 2000, pp.529-540.

[4] E.M. Marcotte, I.Xenarios, and D.Eisenberg, "Mining Literature for Protein-Protein Interactions", *Bioinformatics*, 2002, 17(4): pp.359–363.

[5] J.D.Kim, T.Ohta, Y.Tateisi, and J.Tsujii, "Genia Corpus – a Semantically Annotated for Bio-Texmining, *Bioinformatics*, 19: pp.180-192.

[6] L.Tanabe and W.J.Wilbur, "Tagging Gene and Protein Names in Full Text Article", *In Proceedings of Association for Computational Linguistics*, 2004, pp.9-13.

[7] N.Christianini and J.Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Method,* Cambridge University Press, 2000.

[8] T.Joachims, "A Probabilistic Analysis of the Rocchio Algorithm With TFIDF for Text Categorization", *In Proceedings of 14th International Conference on Machine Learning*, 1997, pp.143-151.

[9] T.Ono, H.Hishigaki, A.Tanigami, and T.Takagi, "Automated Extraction of Information on Protein-Protein Interactions from the Biological Literature", *Bioinformatics*, 2001, 17(2): pp.155-161.