The Integration of Multiple Feature Representations for Protein Protein Interaction Classification Task

Man Lan[12§], Chew Lim Tan[1]

[1]Department of Computer Science, School of Computing, National University of Singapore

[2]Computer Science and Technology Department, East China Normal University

[§]Corresponding author

Email addresses:

Man Lan: lanman.sg@gmail.com

Chew Lim Tan: tancl@comp.nus.edu.sg

Abstract

Background

In order to extract and retrieve protein protein interaction (PPI) information from text, automatic detecting protein interaction relevant articles for database curation is a crucial step. The vast majority of this research used the “bag-of-words” representation, where each feature corresponds to a single word. For the sake of capturing more information left out from this simple bag-of-word representation, we examined alternative ways to represent text based on advanced natural language techniques, i.e. protein named entities, and biological domain knowledge, i.e. trigger keywords.

Results

These feature representations are evaluated using SVM classifier on the BioCreAtIvE II benchmark corpus. On their own the new representations are not found to produce a significant performance improvement based on the statistical significance tests. On the other hand, the performance achieved by the integration of 70 trigger keywords and 4 protein named entities features is comparable with that achieved by using bag-of-words alone. In addition, the only 4 protein named entities features (4PNE) obtained the best *recall* performance (98.13%).

Conclusions

In general, our work supports that more sophisticated natural language processing (NLP) techniques and more advanced usage of these techniques need to be developed before better text representations can be produced. The feature representations with simple NLP techniques would benefit the real-life detecting system implemented with great efficiency and speed without losing the classification performance and exhaustive curation system.

Background

With the rapid growth of biological and biomedical research, the volume of published biological and biomedical articles is expanding at an increasing rate. Characterizing protein protein interaction (PPI) from sentences is one of the most important biomedical problems, which is crucial to understanding not only the functional role of individual proteins but also the organization of the entire biological processes. In practice, before detecting protein interaction descriptions in sentences, it is necessary to pick out those articles which contain relevant information relative to protein interactions. However, due to the rapid growth of the biomedical literature and the increasing number of newly discovered proteins, it is becoming difficult for the interaction database curators to keep up with the literature by manually detecting and curating protein protein interaction information.

In recent years, the work on biomedical literature classification has interested a lot of researchers in Biology, Computer Science and Linguistics. Although the previous attempts to evaluate text classification in the biomedical domain have been made in the context of the TREC Genomics track [1], the recent BioCreAtIvE II Challenge [2] proposed a specific Protein Interaction Article Sub-task 1 (IAS) focusing on the detection of protein protein interaction relevant articles from PubMed titles and abstracts. Most of the participated teams adopted traditional bag-of-words approach to represent text. No advanced NLP techniques or components but stemming and stop words list were adopted. Even though few teams used POS tagging, shallow parsing and sentence splitting, they achieved worse performance than those who used the simple bag-of-words approach. None of these teams adopted complicated advanced

NLP techniques, such as NER. In this challenge, we (team 57) achieved the best F_1 (0.78[1]) performance using 900 words weighted by *tf.rf* scheme [3].

Even though the simple bag-of-words approach has been widely-used for text representation and performed well in practice, it was criticized for ignoring a great deal of useful information from the original document. Therefore, researchers have also adopted several different ways to represent text for biological text classification, for example, predefined entities and keywords [4], expert-defined rules [5], local patterns [6], etc. However, since these features have been examined on the gene expression information, we are interested to explore multiple features for the specific protein protein interaction information.

For this purpose, in this paper, we investigated multiple feature representations to further improve the text classification performance. Besides the traditional domain-independent bag-of-words approach and term weighting methods, we explored other domain-dependent features, i.e. protein interaction trigger keywords, protein named entities. In addition, the integration of these multiple features for this specific PPI text classification task has been evaluated on the BioCreAtIvE II corpus. To the best of our knowledge, so far no such work has been explored in the PPI task.

Methods

In this BioCreAtIvE PPI IAS subtask, we examined different ways for text representation based on information extraction (e.g. named entities) and domain experts (trigger keywords). We also explored the performance of their integration and

---

[1] This result is achieved after BioCreAtIvE refined the released test corpus by removing 37 relevant and 36 non-relevant abstracts during the post-evaluation period. Since the published released results were evaluated on a smaller data set (677 total instances), they are a bit different with our results (750 total instances) in this paper since BioCreAtIvE II has not published which test abstracts were removed from the initial test collection.

the baseline bag-of-words approach. In addition, to check whether the different feature representations are significantly different from each other, we did the statistical significance tests on these representations.

Data Corpus

The training corpus of BioCreAtIvE II challenge in year 2006 is a collection of abstracts which contains 3536 true positive documents (64.3%) which are relevant for PPI curation and 1959 true negative documents (35.7%) which are not relevant for PPI curation from the two databases, i.e. IntAct and MINT. After the training period, participants received a test data of 750 unlabeled abstracts and had to classify them and submit the test results in one week.

The Porter's stemming was performed to reduce words to their base forms. Stop words (513 stop words), punctuations and numbers were removed. The threshold of the minimal term length is 3 (since many biological keywords contain 3 letters, such as acronym). The resulting vocabulary has 24648 words (terms or features). By using the λ^2 statistics ranking metric for feature selection, the top P= {200, 300, 400, 450, 500, 1000, 1500} features from positive and negative categories were selected from the training set. Since the best performance has been achieved using 900 features (bag-of-words) in our previous experiments based on a through evaluation [3], then we only reported this best result by using 900 features in this paper.

The bag-of-words Approach Feature Representation

The representation that dominates the text classification literature is known as the "bag-of-words". For most bag-of-words representations, each feature corresponds to a single word found in the training corpus, usually with case information and punctuation removed. Often infrequent and frequent words are removed from the

original text. Sometimes a list of *stop words* (functional or connective words that are assumed to have no information content) is also removed.

Typically, there is some attempt to make the features more statistically independent, i.e. removing suffixes from words using a *stemming* algorithm. Stemming has the effect of mapping several morphological forms of words to a common feature (in most cases, the stemmed root may not be a complete word).

Besides feature type, another important issue is term (i.e. feature) weighting. Different features have different importance in a text and thus an important indicator represents how much the feature contributes to the semantics of document. Term weighting methods assign appropriate weights to terms to improve the performance of text categorization. We have earlier proposed a new effective supervised term weighting method *tf.rf* (see [7] for more details), which has been confirmed to perform significantly better than other methods (including *tf.idf* and other supervised term weighting methods) on two widely-used newswire benchmark corpora, i.e. Reuters corpus and 20 Newsgroups corpus, cross different learning methods. Therefore, in this new domain, we examine the results of *tf.rf* method as well.

Protein Named Entities (PNEs)

A very basic observation about bag-of-words representation is that a great deal of the information from the original document is discarded. Paragraph, sentence and word order is disrupted, and syntactic structures are also broken. The end result is that the text is represented incoherent to humans in order to make it coherent to a machine learning algorithm. The goal of using protein named entities (PNEs) as features is to attempt to capture some of the information left out of the bag-of-words representation, especially for the specific PPI task.

Recognizing named entities like gene, protein and virus, is quite important for biomedical information retrieval and information extraction. It is a challenging task because there is no standard naming conventions of named entities in the biomedical domain, being much more difficult than the one in the news domain. For example, many biomedical entity names are descriptive and have many words and numbers. One biomedical entity name may be with various spelling forms with capitalization or hyphen or even various irregular abbreviations.

In this paper, we adopted an existing named entity recognition system named PowerBioNE [8], which is based on a Hidden Mixture Markov Model. In this recognition system, various evidential features are integrated through a HMM-based recognizer to deal with various complex naming conventions in the biomedical domain. Due to lack of enough annotated training corpus, we only use PowerBioNE to extract protein names.

In our previous work in [3], we simply considered the existence of PNE in the document as one feature in a text (0 for absence and 1 for presence) and combined PNE with the bag-of-words representation. The previous experiments showed that this combined representation has worse performance than the bag-of-words alone. In consideration of the specific PPI task, our basic idea is that since the PPI articles describe the interaction connections between proteins, there should be more (or at least two) PNEs in the relevant articles. Therefore, unlike our previous work, in this paper, we attempt to use PNE to generate more features in order to capture more information.

The PowerBioNE recognition system has extracted 30780 protein named entities (even more than the 24648 words in the whole resulting vocabulary after stemming and removing stop words) from the training corpus. One noticing phenomenon of

these extracted named entities is the wide distribution in the training and test data set. Table 1 shows the statistics of distribution of abstracts with different number of PNEs in the training and test corpus. Although the accuracy is not very high by using PowerBioNE, there are some issues worthy of discussion. First, it is favourable for relevant articles that vast majority of them (99.1%) have at least two PNEs. Second, 76.68% of non-relevant articles unfavourably have at least two PNEs. This indicates that detecting these non-relevant articles is quite challenging, that is, although these articles are not relevant to PPI task, their contents are naturally close to protein-relevant. Third, 96.53% of test instances have at least two PNEs while only half of test instances are non-relevant. This shows that these test articles are quite noisy and it is more difficult for curators to detecting whether they are relevant.

Another noticing phenomenon is sparse occurrence. Most of the extracted protein named entities occur only once or few times in the corpus. For example, 25740 named entities (83.7%) occur only once, 2529 entities (8.2%) occur more than three times and only 380 entities (1.2%) occur more than ten times in the whole corpus. This sparse occurrence problem makes the document indexing difficult since many documents will be represented as null vectors when the number of named entities used for indexing is quite small. Therefore, we adopted the following four PNE features for representation to avoid the null vectors: (1) if the article has no PNE; (2) if the article has at least one PNE; (3) if the article has at least two PNEs; (4) if the article has more than two PNEs (for each feature, 0 for NO and 1 for YES). For example, for article with PubMed id 1321290, PowerBioNE extracted three PNEs, i.e. p53, e6 and hpv-16, thus it is represented as [0 1 1 1] in this 4-PNE representation.

Trigger Keywords

Table 2 lists 70 stemmed trigger keywords. These trigger keywords are selected out by the biological domain experts based on a great number of PPI documents rather than this BioCreAtIvE corpus alone. These stemmed trigger keywords are selected for several reasons. First, verb trigger keywords express existence and action of proteins and their interactions, which is based on the consideration that relevant PPI abstracts describe interaction events between proteins. Second, noun trigger keywords express the occurrence and locales of proteins and their interactions. Generally, these trigger keywords have been expected to serve as a complement to PNE representation and preserve more information neglected by using PNE feature alone.

Support Vector Machines

Support vector machine (SVM) is a relatively new machine learning algorithm based on the *structural risk minmization* principle from computational learning theory, which seeks, among all the surfaces in $|W|$-dimensional ($|W|$ is the number of features) space that separate the training data examples into two classes, the surface (*decision surfaces*) that separates the positives from the negatives by the widest possible margin. Thus this best decision surface is determined by only a small set of training examples, known as *support vectors*. This quite interesting property makes SVM theoretically unique and different from many other methods, such as *k*NN, Neural Network and Naïve Bayes where all the data examples in the training data set are used to optimize the decision surface [9].

In recent years, SVM has been extensively used in text classification and has been confirmed to show better performance than other conventional machine learning algorithms to handle relatively high dimensional and large-scale training set, see [9] , [10], [11], and [12]. Specially, our benchmark adopted the linear SVM rather than non-linear SVM. The reasons why we chose linear kernel function of SVM in our

experiments are listed as follows. First, linear SVM is simple and fast [11]. Second, linear SVM performs better than the non-linear models [9] and [11]. The SVM software we used is LIBSVM-2.8 [13].

Performance Evaluation

Classification effectiveness is usually measured by using *precision* (*P*) and *recall* (*R*). *Precision* is the proportion of truly positive examples labeled positive by the system that were truly positive and *recall* is the proportion of truly positive examples that were labeled positive by the system. Neither *precision* nor *recall* makes sense in isolation from each other as it is well known from the information retrieval practice that higher levels of *precision* may be obtained at the price of low values of *recall*. Thus, a classifier should thus be evaluated by means of a measure which combines *precision* and *recall*. The most widely-used measures adopted by text classification are F_1 function which attributes equal importance to *precision* and *recall*. Thus, the F_1 function is computed as:

$$F_1 = \frac{2 * P * R}{P + R}$$

Statistical Significance Tests

To compare the performance between two text representations, we employed the McNemar's significance tests [14] based on the micro-averaged F_1 value. McNemar's test is a λ^2-based significance test for goodness of fit that compares the distribution of counts expected under the null hypothesis to the observed counts. Two classifiers f_A and f_B based on two different text representations were performed on the test set. For each example in test set, we recorded how it was classified and constructed the following contingency table (Table 3).

The null hypothesis for the significance test states that on the test set, two classifiers f_A and f_B will have the same error rate, which means that $n10 = n01$. Then the statistic λ is defined as

$$\lambda = \frac{(|n01 - n10| - 1)^2}{n01 + n10}$$

where $n10$ and $n01$ are defined in Table 1.

Dietterich showed that under the null hypothesis, λ is approximately distributed as λ^2 distribution with *1* degree of freedom, where the significance levels 0.01 and 0.001 corresponded to the two thresholds $\lambda_0 = 6.64$ and $\lambda_1 = 10.83$ respectively. Given a λ score computed based on the performance of a pair of classifiers f_A and f_B, we compared λ with threshold values λ_0 and λ_1 to determine if f_A is superior to f_B at significance levels of 0.01 and 0.001 respectively. If the null hypothesis is correct, then the probability that this quantity is greater than 6.64 is less than 0.01. Otherwise we may reject the null hypothesis in favour of the hypothesis that the two text representations have different performance when trained on the particular training set.

Results and Discussion

Table 4 lists the detailed results of different feature representations, where 900BOW means using 900 words (bag-of-words), 4-PNE means using 4 PNEs (see Methods), 70trigger means using 70 trigger keywords. Their combinations are denoted by using "+" sign. For most of each representation, we also tried two different term weighting methods, i.e. the *binary* and *tr.rf* methods. Besides the first 9 runs, we also adopted a simple majority voting technique to further improve the system performance. The results are shown in Table 4 as Run 10 and Run 11, which simply combine the previous three runs, respectively.

Some interesting observations from Table 4 can be found as follows. First, using 4-PNE representation alone achieves the worst F_1 value among all the feature representations. However, the 4-PNE representation has the highest *recall* value among all the feature representations, i.e. **98.13%**. In some real-life scenarios where the underlying end user demands do not focus on the F_1 value only, for example in case of exhaustive curation, a high *recall* might be more desirable. Thus, the 4-PNE representation would be favorable since it only uses 4 features to represent all the articles and consequently it is quite efficient for the on-line curation system.

Second, the bag-of-words approach has the best F_1 performance. Although the *tf.rf* weighting method has higher performance than the *binary* method, the following statistical significance test result shows that the *tf.rf* method is superior to the *binary* method only at significance levels of 0.01 while at significance level of 0.001 there is no significant difference between them.

Third, interestingly, the trigger keywords representation method has much less features (70 words) than the bag-of-words approach (900 words) but it achieved comparable performance with the bag-of-words representation. This observation is interesting that in real-life application, the detecting system will benefit from less features and faster indexing and predicting process. However, when combined with 4-PNE representation, the 70 trigger keywords representation has not made any significant improvement. This result is beyond our original expectation that this combination of trigger keywords and PNE would capture more information than the bag-of-words approach or 70 trigger keywords or 4-PNE alone.

In addition, we also examined the performance of the integration of the above different features. Whether using *tf.rf* weighting method or the *binary* method, this

integration has comparable results with the bag-of-words approach. The statistical significance tests indicated that there is no significant difference between them.

Finally, we also adopted a simple majority voting technique in order to further improve the performance. However, the significance tests also indicated that the majority voting technique either has no significant improvements (Run 11) or even decrease the performance of the bag-of-words approach (Run 10).

Based on the experimental results, we state that extraction useful information from sentence level is necessary. For example, from abstract level, many negative abstracts contain protein named entities and trigger keywords in the content even though they are not relevant to PPI information. In most cases, these PNEs and/or trigger keywords are in different sentences and there is no interaction connection between these PNEs. On the contrary, although most PPI relevant articles have favourable PNEs and trigger keywords, some may not contain any extracted PNE or protein connection keywords. Therefore, in order to capture the true meaning of abstracts rather than the PNE or trigger keywords alone, we need to get into the sentence level and find out more useful feature representations by using advanced NLP techniques in our future work.

Conclusions

In this paper, we examined multiple feature representations for protein protein interaction classification task, i.e. bag-of-words approach with different term weighting schemes, protein named entities, trigger keywords and their combination. Although the statistical significance tests showed that there is no significant difference between most of them, the feature representations with simple NLP techniques have some interesting results. For example, the only 4 protein named entities features (4-PNE) obtained the best *recall* performance and which can be

adopted for specific end users' demands for exhaustive curation. In addition, the 70 trigger keywords representation achieved the comparable good performance with the bag-of-words approach, which benefits the real time detecting system implemented with great efficiency and speed demands due to a smaller feature space with less dimensions.

Acknowledgements

We would like to thank Dr. Su Jian from the Institute of Infocomm Research, Singapore for discussion and suggestions on the relevant work.

References

1. W. Hersh, A. Cohen, J. Yang, R.T. Bhupatiraju, P. Roberts, M. Hearst: **TREC 2005 Genomics Track Overview**. *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)* 2005.
2. BioCreAtIvE [http://biocreative.sourceforge.net/].
3. Man Lan, Chew Lim Tan and Jian Su: **A Term Investigation and Majority Voting for Protein Interaction Article Sub-task 1 (IAS)**. In *the Proceedings of the Second BioCreative Challenge Evaluation Workshop*. 2007.
4. Keerthi, S. S., Ong, C. J., Siah, K. B., Lim, D. B., Chu, W., Shi, M., Edwin, D. S., Menon, R., Shen, L., Lim, J. Y., and Loh, H. T: **A machine learning approach for the curation of biomedical literature: KDD Cup 2002 (task 1)**. *SIGKDD Explor. Newsl.* 4, 2 (Dec. 2002), 93-94.
5. Regev, Y., Finkelstein-Landau, M., Feldman, R., Gorodetsky, M., Zheng, X., Levy, S., Charlab, R., Lawrence, C., Lippert, R. A., Zhang, Q., and Shatkay, H: **Rule-based extraction of experimental evidence in the biomedical domain: the KDD Cup 2002 (task 1)**. *SIGKDD Explor. Newsl.* 4, 2 (Dec. 2002), 90-92.

6. Ghanem, M. M., Guo, Y., Lodhi, H., and Zhang, Y: **Automatic scientific text classification using local patterns: KDD CUP 2002 (task 1)**. *SIGKDD Explor. Newsl.* 4, 2 (Dec. 2002), 95-96.

7. Man Lan, Chew Lim Tan and Hwee Boon Low: **Proposing a New Term Weighting Scheme for Text Categorization**. In *the Proceedings of the 21st National Conference on Artificial Intelligence (AAAI '06),* 2006.

8. GuoDong Zhou and Jian Su: **Exploring deep knowledge resources in biomedical name recognization**. In *the Proceedings of JNLPBA shared task*, 99-102, 2004.

9. Yiming Yang and Xin Liu: **A re-examination of text categorization methods**. In *the proceedings of the 22nd annual international ACM SIGIR conference*, 42-49, New York, 1999.

10. Thorsten Joachims: **Text categorization with support vector machines: learning with many relevant features**. In *the proceedings of ECML-98*, 137-142, Chemnitz, DE, 1998.

11. Susan Dumais, John Platt, David Heckerman, and Mehran Sahami: **Inductive learning algorithms and representations for text categorization**. In *proceedings of the 7th CIKM*, 148-155, 1998.

12. Edda Leopold and Jorg Kindermann: **Text categorization with support vector machines. how to represent texts in input space?** *Machine Learning*, 46(1-3):423-444, 2002.

13. Chih-Chung Chang and Chih-Jen Lin: **LIBSVM: a library for support vector machines**, 2001. http://www.csie.ntu.edu.tw/ cjlin/libsvm.

14. Thomas G. Dietterich: **Approximate statistical tests for comparing supervised classification learning algorithms**. *Neural Comput.*, 10(7) : 1895-1923, 1998.

Tables

Table 1 - Distribution of abstracts with different number of Protein Named Entities (PNE) in the training and test corpus

Data set	***Sum***	***#_no_PNE***	***#_one_PNE***	***#_two_PNEs***	***#_more_PNEs***
Relevant training	3536	11 (0.31%)	21 (0.59%)	47 (1.33%)	3457 (97.77%)
Non-relevant training	1959	266 (13.58%)	191 (9.75%)	206 (10.52%)	1296 (66.16%)
test	750	13 (1.73%)	13 (1.73%)	36 (4.80%)	688 (91.73%)

Table 2 - 70 Stemmed Trigger Keywords List

accumul	complex	express	interfac	phosphorylat	repress
activ	contain	impair	intra	produc	residu
add	decreas	inact	involv	product	secret
addit	demethyl	inactiv	mediat	promot	sever
addition	dephosphoryl	increas	methylat	protein	stimul
apoptosi	deplet	induc	modif	react	substitut
associ	disassembl	induct	modifi	reduc	surfac
bind	discharg	influenc	modul	reduct	transactiv
block	domain	inhibit	myogenesi	regul	upregul
bound	downregul	initi	overexpress	regulat	up-regul

catalyz	down-regul	inter	particip	releas
cleav	elev	interact	phosphoryl	replac

Table 3 - McNemar's Test Contingency Table

$n00$: Number of examples misclassified by both classifiers f_A and f_B	$n01$: Number of examples misclassified by f_A but not by f_B
$n10$: Number of examples misclassified by f_B but not by f_A	$n11$: Number of examples misclassified by neither f_A nor f_B

Table 4 - Detailed Performance of multiple feature representations

Run	***Representation***	***Weighting***	***Precision***	***Recall***	***F_1***	***Accuracy***
1	900BOW	(*binary*)	67.32	81.87	73.89	71.07
2	900BOW	(*tf.rf*)	**69.59**	86.67	**77.20**	**74.40**
3	4PNE	(*binary*)	53.49	**98.13**	69.24	56.40
4	70Trigger	(*binary*)	67.41	80.53	73.39	70.80
5	70Trigger	(*tf.rf*)	66.81	83.73	74.32	71.07
6	70Trigger+4PNE	(*binary*)	67.76	82.40	74.37	71.60
7	70Trigger+4PNE	(*tf.rf*)	67.79	85.87	75.76	72.53
8	BOW+Trigger+PNE	(*binary*)	67.69	82.13	74.22	71.47
9	BOW+Trigger+PNE	(*tf.rf*)	69.21	86.93	77.07	74.13
	Majority Voting					
10	Run: 3+6+8		58.48	86.40	69.75	62.53
11	Run: 3+5+2		65.28	92.27	76.46	71.60