Text Classification using Ontology and Semantic Values of Terms for Mining Protein Interactions and Mutations

Berna Altınel [1], Zehra Melce Hüsünbeyi [2] and Arzucan Özgür [3]

[1] Department of Computer Engineering, Marmara University, TR-34722, Kadıköy, Istanbul, Turkey

{berna.altinel}@marmara.edu.tr

[2,3] Department of Computer Engineering, Boğaziçi University, TR-34342, Bebek, Istanbul, Turkey

{arzucan.ozgur,zehra.husunbeyi}@boun.edu.tr

Abstract*—There is a big amount of published articles in the biomedical domain. Automatically processing and extracting biologically crucial information such as protein-protein interactions from the biomedical literature is one of the main difficulties for both research and commercial platforms. We contributed to the Document Triage Task of the Precision Medicine Track of the BioCreative VI challenge assessment by developing text mining methods to assist health professionals and researchers. We build three methods, which are capable of receiving a list of PMIDs and returning a relevance-ranked judgement for the articles in order to identify relevant PubMed articles, which are about genetic mutations affecting protein-protein interactions. Our first approach is based on meaning calculation, which computes the words' meaning scores in the scope of classes. Meaning computation is based on the Helmholtz principle and has been utilized for various applications in the field of text mining like feature extraction, information retrieval, text classification, and text summarization. Nevertheless, to the best of our knowledge, our effort is the first work, which uses meaning calculation of the terms retrieved from the Interaction Network Ontology (INO) to construct a semantic classifier. Our second approach depends on the extraction of the most salient terms, generated from Genia Tagger[1], taking advantages of the term frequency-relevance frequency (TF-RF) metric. This methodology also uses sprinkling, which is a process of adding further terms corresponding to class labels of documents to the training documents in order to strengthen class-based relationships in the training phase. Our third approach is based on using a Convolutional Neural Network (CNN). In our model, the first layer embeds words into low-dimensional vectors. Convolutions over the embedded word vectors are performed in the next layer by using multiple filter sizes. The next stage is max-pooling the result of the convolutional layer into a long feature vector. Finally, the results are classified with the softmax layer. In order to evaluate our results, we also implemented linear kernel with Support Vector Machines (SVM) and Naïve Bayes (NB) as the baseline algorithms. According to the experiment results the presented methods outperform the baseline algorithms.

Keywords— text mining, meaning, TF-RF, sprinkling, CNN.

[1] http://www.nactem.ac.uk/GENIA/tagger/

I. INTRODUCTION

The huge volume of published articles in the scientific literature continues to enlarge by the contributions of millions of people every day. It is vitally important to extract clinically useful information that links genes, mutations, and diseases to specialized treatments from this published literature for the precision medicine initiative (PMI), which aims to find individualized treatment for a patient according to his/her genetic profile.

Protein-protein interactions (PPIs) are crucial for a range of biological processes such as cell cycle control, DNA replication, signal transduction etc. Mutations may affect the stability and affinity of protein-protein interactions. Thus, combining the efforts in protein-protein interaction [1, 2] and mutation extraction [3] has high significance for precision medicine.

Donaldson et al. [4] present two classifiers for finding protein-protein interaction data in PubMed. According to their experiments performed on the BIND database [5], the SVM method with linear kernel is reported to give higher classification results than a Naïve-Bayesian classifier. Mitsumori et al. [6] present one of the other systems where SVM is used. They use the bag-of-words (BOW) feature representation for the words that are closest to the protein names in order to get protein-protein interactions. They compare their own system's capability on extracting protein-protein interactions, to that of other systems' in the literature. A semi-supervised information extraction approach for identifying sentences in text that show an interaction relation between two proteins is suggested in [7]. This methodology depends on the analysis of the paths between two protein names in the dependency parse trees of the sentences. They mention that they get significant improvement over the results of the existing methods in the literature. Participants in the Protein-Protein Interaction tasks of the BioCreative II and BioCreative III Challenges also attempt to develop methods that aim to detect interaction relevant articles by using numerous techniques. Alex et al. [8] apply a SVM classifier

with the usage of pre-processing, part-of-speech (POS) tagging, sentence splitting and shallow parsing. They report 70% precision, 86% recall and 77% F-score. Another study present a methodology which integrates protein name detection and abbreviation resolution systems by also using SVM and obtain 78% F- score [9].

In this paper, we present our participation in the Document Triage Task of PrecMed Track of the BioCreative VI Challenge assessment by developing text-mining methods to assist health professionals and researchers. We present three methods for identifying the articles about genetic mutations affecting protein-protein interactions. Our approaches are based on class-based semantic values of terms, ontology and convolutional neural networks.

II. RELATED WORK

Term Frequency-Relevance Frequency (TF-RF)

Term Frequency-Relative Frequency (TF-RF) is presented by Lan et al. [9] and is a supervised term weighting method, which is based on the number of positive and negative documents that a term occurs in. In a text classification setting a selected category is labeled as the positive category, while all other categories in the same dataset are labeled as the negative category. The formula of TF-RF is:

$$TF-RF = tf_w \times \log\left(2 + \frac{a}{\max(1,c)}\right) \quad (1)$$

where tf_w shows the term frequency of word *w*, *a* denotes the number of documents in the positive category which include word *w*, and *c* is the number of documents in the negative category which include word *w*. According to an explanatory example given in [9]; in contrast to IDF, with RF methodology each word is assigned more appropriate weights from the point of different categories since RF considers the category information.

Helmholtz Principle from Gestalt Theory and Meaning Calculation

According to the Helmholtz principle in human perception from Gestalt Theory, humans easily notice events with a large deviation from noise or randomness [10]. A number of explanatory examples are given by Balinsky et al. [10] and these examples show that interesting events and meaningful features appear in great deviations from randomness.

Textual data comprise structures like sentences, paragraphs and documents. Balinsky et al. [10] try to describe the meaningfulness of these structures by utilizing the Helmholtz principle. A meaning value is given to each word for modeling the meaningfulness of these structures. Balinsky et al. [10] mention that a sharp rise in frequencies can be used in quick modification discovery. A burst is a period of increased and quick modifications in an event as mentioned in [11].

The meaning value of a term (word) *w* in a class c_j is computed with Eq. (2) [10]:

$$meaning(w, c_j) = -\frac{1}{m}\log\binom{K}{m} - \left[(m-1)\log N\right] \quad (2)$$

where *w* denotes a word, *m* shows the occurrence of term *w* in class c_j, *K* specifies the frequency of term *w* in the whole dataset. *N=L/B; L* represents the length of the dataset and *B* represents the length of the class c_j in number of terms [12]. If a word's meaning score in a specific class is larger, then this means that this word is more informative for that class. A meaning value of a word essentially shows how high this word's frequency is likely to be in a class of documents compared to the other classes of documents.

Sprinkling

Sprinkling is a process of adding further terms corresponding to class labels of documents to the training documents in order to strengthen class-based relationships in the training phase. For instance, in [13] Latent Semantic Indexing (LSI) is performed both on standard term-document matrix and term-documents matrix augmented with sprinkled terms. The sprinkling process is shown in Figure 1:

	t_1	t_2	t_3	…	t_j
d_1	1	0	1		1
d_2	0	1	1		0
d_3	1	1	1		1
…					
di	1	1	0		0

Sprinkling →

	t_1	t_2	t_3	…t_j	$\mathbf{c_1}$	$\mathbf{c_2}$	$\mathbf{c_3}$	…$\mathbf{c_n}$
d_1	1	0	1	1	**1**	**0**	**0**	**0**
d_2	0	1	1	0	**0**	**1**	**0**	**0**
d_3	1	1	1	1	**1**	**0**	**0**	**0**
…								
d_i	1	1	0	0	**0**	**1**	**0**	**0**
…								
d_r	0	0	0	1	**0**	**0**	**1**	**0**

(a) Original term-document matrix **(b)** Sprinkled term-document matrix

Figure 1. (a) Original term-document matrix with *r* documents and *j* terms. **(b)** term-document matrix after sprinkling with *n* terms. These new additional terms show the class labels of the corresponding documents. For instance, d_1 belongs to class c_1, d_2 belongs to class c_2, d_3 belongs to class c_1, d_i belongs to class c_2…. etc.

Chakraborti et al. [13] drop sprinkling terms after performing LSI. Test documents are classified using k-Nearest Neighbors (kNN) with the Euclidean distance metric. According to their experimental results the presence of sprinkling terms improves the classification performance. For instance, the classification accuracies of Sprinkled-LSI on four different subgroups of 20NewsGroups[1] dataset are reported as 86.99%, 80.60%, 80.42%, and 93.89% while the classification accuracies of LSI are reported as 79.32%, 72.55%, 66.30% and 91.17%; respectively [13]. They state that the integration of further knowledge, which represents the latent class structure, improves the classification performance.

[1] http://qwone.com/~jason/20Newsgroups/

Consequently; with sprinkling process, documents related to the same class are located closer to each other.

Convolutional Neural Network (CNN)

Convolutional neural networks apply layers with convolving filters, which are used as local features [17]. CNN models have achieved a notable performance in the natural language processing field, particularly in the task of sentence classification [14, 15, 16).

A CNN architecture with multiple convolution layers was proposed by Kalchbrenner [15]. This model consists of positing latent, dense and low-dimensional word vectors (initialized to random values) as inputs.

Kim's [14] model, shown in figure 2, proved that CNN can obtain state-of-the-art results with a simple one layer architecture. This model uses pre-trained word vectors as inputs. This is followed by a convolution and maxpooling layer, and a softmax classifier.

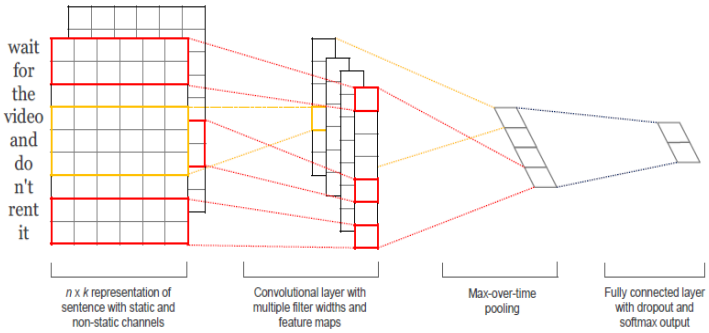

Figure 2. Model architecture with two channels for an example sentence [14].

A similar model is proposed by Johnson and Zhang [16], but they used high dimensional 'one-hot' vector representations of words as CNN inputs. Their focus was on classification of longer texts, rather than sentences.

III. METHODS

A. *Semantic Meaning Classifier with INO (SMC-INO)*

In a novel text classification algorithm named Supervised Meaning Classifier (SMC) [18], the authors calculate the meaning scores of the words in a document for a certain category and sum them to obtain a relative class membership value of the document for that category. In other words, the class membership of a particular document is determined by the sum of the meaning or the importance of its terms for that particular class. This is somewhat similar to the Naive Bayes algorithm where the class conditional document probability $P(D|C)$ is calculated by multiplying probabilities of the class conditional term probabilities $P(w|C)$ in addition to a class prior probability $P(C)$. The SMC classifier uses meaning calculations as explained in Section II. Ganiz et al. [18] calculate the meaning scores of each word in the training set for each class, which constitutes the training phase. In the classification phase, for an unlabeled new test instance, meaning scores of the words for a particular class are summed up to obtain class membership value. The class with largest membership value is chosen as the label of the instance. SMC is show to be superior to Multinomial NB and SVM with linear kernel, especially on inadequate training data [18].

In our experiment setting, we first find interaction terms by using the Interaction Network Ontology (INO) [19] and generate training and test datasets according to these terms. Then, we use the SMC algorithm for classifying the documents.

B. *Sprinkled Relevance Value Classifier (S-RVC)*

We implemented a Relevance Value Classifier similar to SMC with the only difference that the class-based term values are calculated by using the TF-RF metric described in Section II. We developed Sprinkled Relevance Value Classifier (S-RVC) which has the same architecture with RVC with the only difference that; S-RVC uses additional terms which represent the class relationships between documents. In other words, class labels are added into the standard term-document matrix in order to enrich the class knowledge in the training corpus and add this information into the classification model.

C. *Convolutional Neural Network (CNN)*

The CNN model architecture, which is related to Kim Yoon's model [14], includes several layers such as embedding, convolution, max-pooling, dropout and softmax. In the first layer that defines the embedding layer, vocabulary word indices correspond to low-dimensional vectors and embeddings are learned from scratch. Then convolutions are performed on the embedding matrix via linear filters of different sizes (3, 4, 5). Each filter operates for generating a feature map for a window of words. This is followed by a max-pooling operation [16], which is applied, to each feature mapping for inducing a particular feature vector. For regularization, the dropout method is used which avoids co-adaptations of hidden units to decrease overfitting. Finally, with the softmax layer normalized probabilities over labels are generated. In our experiments, the hyperparameters, which are the number and sizes of convolutional filters, dropout rate, and mini-batch size, are chosen with investigating their impacts on performance during the training phase.

IV. EXPERIMENT SETUP

A. *Data set*

The training dataset of Biocreative VI Document Triage Task has 4082 PubMed articles. These articles are manually labelled as relevant (i.e. describing genetic mutations affecting protein-protein interactions) or not relevant by BioGRID database curators. This dataset has skewed class distribution, since 1729 of these documents are labelled as relevant while 2353 of them are labelled as not relevant. There are 17765 words in the training dataset of Biocreative VI Document Triage Task. The provided training set is further split into training (70%) and test (30%) sets for evaluation purposes as shown in Table 1.

Table 1. Overview of training and test splits of the dataset

Dataset	# Relevant citations	#Not Relevant citations	#total citations
Training	1210	1769	2979
Test	519	584	1103

Interaction Network Ontology (INO) and Genia Tagger

The Interaction Network Ontology [19] was used to select protein related interaction keywords from the literature. We obtained a sequence of literature mining keywords common in the training dataset, which includes 437 tokens. Named entity tags, which are cell lines, cell types, DNA, RNA and protein names, are extracted from the training dataset with the Genia Tagger [20]. In our experiment settings, we used the detected terms via INO and Genia Tagger to increase the performance of the document classification algorithms.

B. Experiment Setting and Evaluation

We apply stemming and stopword filtering to these datasets. After running the algorithms on 10 random training and test splits, we report the average of these 10 results. The evaluation metrics in our experiments are precision/recall and F-Measure.

C. Baseline Algorithms

We use two baseline algorithms in order to evaluate the results of our methods. The first baseline algorithm is the customary linear kernel of SVM. The secondly baseline algorithm is Naïve Bayes.

V. RESULTS AND DISCUSSION

We evaluate and compare the three proposed methods with the two baseline algorithms, which are commonly used high accuracy techniques in the text classification field. The experiment results are shown in Table 2. According to Table 2, the F-scores of the baseline algorithms are 60.3% and 61.4%, for SVM with linear kernel and Naive Bayes, respectively. It is important to note that linear kernel is the traditional state of the art algorithm in SVM for text classificationin domain [21,22]. We achieve our best F-score performance of 88.6% with CNN, which outperforms the two baseline algorithms with a significant difference. While SVM with linear kernel achieves the highest precision of 92.0%, it performs remarkably worse than all other algorithms in terms of F-score. Furhermore, our other two algorithms, S-RVC, SMC-INO, obtain 75.9% and 86.2% F-scores as shown in Table 2. The algorithm which gets the lowest F-score among our algorithms is S-RVC; still it obtains higher F-score in comparison to both linear kernel and Naive Bayes. The superioty of S-RVC over both baseline algorithms could be explained with the usage of class-based relavence values of terms and additional sprinkled terms. Similarly, the superiority of SMC-INO over both baseline algorithms could be explained with the usage of class-based meaning values with the integration of terms retrieved from INO. The advantages of semantic text classification over the traditional text classification are analyzed and discussed in several studies in the literature such as [23, 24, 25, 26]. On the other hand, we get the highest F-score performance over the baseline algorithms with CNN. This may not be surprising, since the high performance of deep learning algorithms in comparison to other existing algorithms are presented in many recent studies [27, 28, 29; 30].

Table 2. Experiment results

Method	***Precision***	***Recall***	***F-Measure***
linear kernel	0.920	0.437	0.603
Naive Bayes	0.832	0.461	0.614
SMC-INO	0.774	0.921	**0.862**
S-RVC	0.755	0.745	**0.759**
CNN	0.863	0.912	**0.886**

Furthermore, SMC-INO, S-RVC and CNN obtain the following precision, recall and F1 values on the real test dataset of the Document Triage Task of the Precision Medicine Track of the BioCreative VI challenge: SMC-INO obtain 0.4886 average precision, 0.5849 recall and 0.5268 F1; S-RVC obtain 0.5055 average precision, 0.7178 recall and 0.5865 F1;CNN obtain 0.5098 average precision, 0.9795 recall and 0.6685 F1.

VI. CONCLUSION AND FUTURE DIRECTIONS

We contributed to the Document Triage Task of PrecMed Track of the BioCreative VI challenge assessment by presenting three methods identifying PubMed articles which are relevant to genetic mutations affecting protein-protein interactions. We also implement two baseline algorithms. We achieve our best F-score performance of 88.6% with CNN, which outperforms the two baseline algorithms with a large margin. Our experimental results show the promise of our novel techniques, S-RVC and SMC-INO. To the best of our knowledge, our's is the first attempt to build these approaches and apply them in the BioNLP domain. Moreover, CNN shows remarkable superiority over the baseline algorithms and it forms a foundation that is open to several improvements. As future work, we would like to analyze and shed light on how our new approaches and CNN implicitly capture semantic information in the context of a class when calculating the similarity between two documents. We also would like to develop a semantic kernel and compare it with the traditional linear kernel. In addition, we plan to implement an algorithm for the relation extraction Task of the PrecMed Track of the BioCreative VI challenge assessment and further improve our approaches.

ACKNOWLEDGMENT

We would like to thank the BioCreative Task organizers for organizing the shared task and for their help with the data preparation and the questions.

REFERENCES

1.Kim S, Islamaj Dogan R, Chatr-Aryamontri A, Chang CS, Oughtred R, Rust J, et al.(2016). BioCreative V BioC track overview: collaborative biocurator assistant task for BioGRID. Database : the journal of biological databases and curation.

2.Krallinger M, Vazquez M, Leitner F, Salgado D, Chatr-Aryamontri A, Winter A, et al. The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. BMC bioinformatics. 2011;12 Suppl 8:S3.

3. Wei CH, Harris BR, Kao HY, Lu Z. tmVar: a text mining approach for extracting sequence variants in biomedical literature. Bioinformatics. 2013;29(11):1433-9.

4. Donaldson, J. Martin, B. de Bruijn, C. Wolting, V. Lay, B. Tuekam, S. Zhang, B. Baskin, G. D. Bader, K. Michalockova, T. Pawson, and C. W. V. Hogue. (2003) Prebind and textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. BMC Bioinformatics, 4:11.

5. Bader, G. D., Betel, D., & Hogue, C. W. (2003). BIND: the biomolecular interaction network database. Nucleic acids research, 31(1), 248-250.

6. Mitsumori, T., Murata, M., Fukuda, Y., Doi, K., & Doi, H. (2006). Extracting protein-protein interaction information from biomedical text with SVM. IEICE Transactions on Information and Systems, 89(8), 2464-2466.

7. Erkan, G., Özgür, A., & Radev, D. R. (2007, June). Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In EMNLP-CoNLL (Vol. 7, pp. 228-237).

8. Alex, B., Grover, C., Haddow, B., Kabadjov, M., Klein, E., Matthews, M., ... & Wang, X. (2008). Automating curation using a natural language processing pipeline. Genome Biology, 9(2), S10.

9. Lan M, Tan CL, Su J, Lu Y. Supervised and traditional term weighting methods for automatic text categorization. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2009;31(4):721-735.

10. Balinsky, A., Balinsky, H., Simske, S., 2011a. On the Helmholtz Principle for Data Mining. In: Proceedings of Conference on Knowledge Discovery, Chengdu, China.

11. Kleinberg, J., 2002. Bursty and Hierarchical Structure in Streams, Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 7(4):373-397.

12. Balinsky, A., Balinsky, H., Simske, S., 2011b. Rapid change Detection and Text Mining. In: Proceedings of 2nd Conference on Mathematics in Defense (IMA), Defense Academy, UK.

13. Chakraborti, S., Lothian, R., Wiratunga, N., & Watt, S., 2006. Sprinkling: supervised latent semantic indexing. In European Conference on Information Retrieval (pp. 510-514). Springer Berlin Heidelberg.

14. Kim, Y., 2014. Convolutional Neural Networks for Sentence Classification, CoRR.

15. Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

16. Johnson, R., & Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv:1412.1058.

17. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner. 1998. Gradient-based learning applied to document recognition. In Proceedings of the IEEE, 86(11):2278–2324, November.

18. Ganiz, M. C., Tutkan, M., & Akyokus, S., 2015. A novel classifier based on meaning for text classification. In Innovations in Intelligent SysTems and Applications (INISTA), 2015 International Symposium on (pp. 1-5). IEEE.

19. Özgür, A., Hur, J., & He, Y. (2016). The Interaction Network Ontology-supported modeling and mining of complex interactions represented with multiple keywords in biomedical literature. BioData mining, 9(1), 41.

20. Tsuruoka, Y., Tateishi, Y., Kim, J. D., Ohta, T., McNaught, J., Ananiadou, S., & Tsujii, J. I. (2005, November). Developing a robust part-of-speech tagger for biomedical text. In Panhellenic Conference on Informatics (pp. 382-392). Springer, Berlin, Heidelberg.

21. Joachims, T., 1998. Text Categorization With Support Vector Machines: Learning With Many Relevant Features, pp.137-142, Springer Berlin Heidelberg.

22. Dumais, S., Platt, J., Heckerman, D., Sahami, M., 1998. Inductive Learning Algorithms And Representations For Text Categorization. In: Proceedings of the Seventh International Conference on Information Retrieval and Knowledge Management (ACM-CIKM), pp. 148–155.

23. Bloehdorn, S., Moschitti, A., 2007. Combined Syntactic and Semantic Kernels for Text Classification, Springer, 307-318.

24. Siolas, G., d'Alché-Buc, F., 2000. Support Vector Machines Based On a Semantic Kernel for Text Categorization, Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, 5(1):205-209.

25. Wang, P., Domeniconi, C., 2008. Building Semantic Kernels for Text Classification Using Wikipedia, Proceeding of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 713-721.

26. Kim, K., Chung, B. S., Choi, Y., Lee, S., Jung, J. Y., & Park, J., 2014. Language independent semantic kernels for short-text classification. Expert Systems with Applications, 41(2), 735-743.

27. Cao, Z., Li, S., Liu, Y., Li, W., & Ji, H., 2015. A Novel Neural Topic Model and Its Supervised Extension. In AAAI (pp. 2210-2216).

28. Hill, F., and Korhonen, A., 2014. Learning abstract concept embeddings from multi-modal data: Since you probably can't see what I mean. In Proceedings of EMNLP, 255–265.

29. Hinton, G., Salakhutdinov, R., 2011. Discovering binary codes for documents by learning deep generative models. Topics Cogn Sci 3(1):74–91.

30. Le, Q. V., & Mikolov, T., 2014. Distributed representations of sentences and documents. arXiv preprint arXiv:1405.4053.