MINING ADVERSE DRUG REACTION MENTIONS IN TWITTER WITH WORD EMBEDDINGS

WENTING WANG
InnoCellence Systems
Email: wenting.wang@innocellence.com

This paper describes our system used in the PSB 2016 Workshop on Social Mining Shared Task for adverse drug reaction (ADR) extraction in Twitter. Our system uses Conditional Random Fields to train a classifier for extracting ADR mentions. We leverage word representations from large amount of unlabeled tweets, both drug related and generic. Our experiment results show that cluster features derived from word representations significantly improve Twitter ADR performances.

1. Introduction

Post-market drug safety surveillance is crucial in identifying potential adverse drug reaction (ADR) uncovered in clinical trials [1]. The rapidly growing social media have opened new opportunities since users tend to discuss their health-related experiences in such platforms. Mining ADR in tweets has recently received significant interest in pharmacovigilance research [2, 3, 4].

However, user generated content in Twitter is short, noisy, and highly informal. Moreover, medical concepts in tweets are often expressed in nontechnical and descriptive way [5]. All these abovementioned problems present various challenges. The PSB 2016 Workshop on Social Mining Shared Task is organized in response to exploit in applying natural language processing (NLP) techniques for automatically extracting ADRs in tweets.

We participated in Subtask 2 of the above Shared Task that aims to identify the span of ADR. We model the problem as a sequential labeling task, using Conditional Random Fields (CRF) as the training algorithm. An additional post-processing step is applied to further refine the output.

The remainder of this paper is structured as follows. In Section 2, we report on the external resources used by our system and how they are obtained and processed. In Section 3, the features used are described in details. In Section 4, the experiment and official results are presented. Finally, Section 5 summarizes our work.

2. External Resources

Our system uses a variety of external resources, either publicly available, or collected and preprocessed by us.

2.1. *ADR Lexicon*

We use the ADR lexicon list compiled by the task organizers.[a] It includes 13,591 phrases from COSTART, SIDER and a subset of CHV, and 136 ADR phrases frequently tagged in training data.

[a] http://diego.asu.edu/downloads/publications/ADRMine/ADR_lexicon.tsv

2.2. *Drug-related word2vec Word Clusters*

We use the pre-trained drug related word clusters generated by the task organizers.[b] It consists of 150 word clusters trained using word2vec.[c]

2.3. *Unlabeled Corpora*

We gather two sets of unlabeled corpora from Twitter: (1) generic raw tweets collected between the period of March 2015 and June 2015, (2) drug-related raw tweets collected using a list of 81 drugs[d] between period of July 2015 and Aug 2015.

The collected raw tweets are tokenized[e] and non-English tweets are removed by following [6], resulting in a total of 68 million generic tweets and 99 thousand drug related tweets.

3. Features

This section describes the features used in our system. Besides the features used in previous work [4], we focus on the use of word cluster features that have shown to be effective in many other NLP tasks [7, 8].

3.1. *POS Feature*

All the tweets are tokenized and POS tagged by Stanford Parser [9]. POS tag of the current word is used as feature.

3.2. *Word Features*

The lowercase format of preprocessed current word is used as feature. For preprocessing, we apply spell correction and stemming. For spell correction, we utilize two resources: (1) for short words (i.e. length <12), we call a python-based Spellchecker;[f] (2) otherwiese, we call the Apache Lucene spell checker library,[g] which suggests the correct spelling based on the ADR lexicon (Section 2.1) and Spell Checker Oriented Word Lists.[h] For stemming, we use RiTa library,[i] which returns the WordNet [10] root of the token. To provide additional context information, three preceding and three following words are also used. This context window size is chosen based on 5-fold cross validation experiment results.

3.3. *ADR Lexicon Feature*

This binary feature shows whether or not the current word exists in the ADR lexicon (Section 2.1).

[b] http://diego.asu.edu/Publications/ADRMine.html
[c] https://code.google.com/p/word2vec
[d] http://diego.asu.edu/downloads/publications/ADRMine/drug_names.txt
[e] https://github.com/myleott/ark-twokenize-py
[f] http://norvig.com/spell-correct.html
[g] http://archive.apache.org/dist/lucene/java/2.9.4/
[h] http://wordlist.aspell.net
[i] https://rednoise.org/rita

3.4. *Negation Feature*

This binary feature indicates whether or not the current word is negated. The negation is identified based on two patterns: negation phrase precedes current token and negation phrase follows current token [11]. Negation phrase is residing up to five words away from current token. The negation pattern is matched to clause level.[j] Feature value for double negation is marked as False.

3.5. *Word Cluster Features*

Besides using the pre-trained drug-related word clusters (Section 2.2), we also use the processed generic and drug-related tweet corpora (Section 2.3) to generate additional K-means clusters and Brown clusters. These additional clusters are constructed by following the steps described in [12], utilizing Brown clustering tool by Percy Liang[k] and Stanford GloVe tool.[l]

We create seven features including the cluster number for current word, three preceding and three following tokens for each cluster file. Then we test a random subset of additional cluster files and select the best cluster file according to 5-fold cross validation performance. Our final settings use generic Brown clusters, pre-trained drug-related clusters, generic and drug-related GloVe clusters.

4. Experiments and Results

Our system is trained using CRFsuite.[n] We encode the ADR mention's boundary with IOB scheme. Therefore, the classifier learns to distinguish between five different labels: *B-ADR*, *I-ADR*, *B-Indication*, *I-Indication* and *Out*.

We also perform a postprocessing step based on heuristic rules to further refine the system output. To prevent false positive, every ADR or Indication mention must contain at least 2 characters and all characters of the mention must be alphabets.

The training corpus including all data from *train*, *devtest* and *test* contains 2,130 tweets where only 1,482 tweets are accessible when the task is held. Among the downloadable tweets, 585 do not contain any annotated ADR or Indication mentions. We make use of all available training data and conduct 5-fold cross validation experiments.

The final evaluation metric for this task is F-measure by comparing the system's output with the gold standard via approximate matching [13].

4.1. *Preliminary Results on Training Data*

Table 1 shows the 5-fold cross validation performances for different feature groups. Baseline feature set includes POS, word (without spell check), ADR lexicon and negation. For simplicity, we use Instance Accuracy defined by CRFsuite to evaluate cross validation performance. The use of a combination of word clusters significantly outperforms the baseline. This demonstrates the usefulness of word vectors in improving the accuracy of a Twitter ADR extraction system.

[j] http://sentiment.christopherpotts.net/lingstruc.html#negation

[k] https://github.com/percyliang/brown-cluster

[l] http://nlp.stanford.edu/projects/glove/

[n] http://www.chokkan.org/software/crfsuite/

Table 1. The effectiveness of cluster features based on 5-fold cross validation.

Feature	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Overall
Baseline	0.5118	0.4747	0.5304	0.4797	0.4189	0.4831
Baseline + Pre-trained drug-related Cluster	**0.5455**	0.5017	0.5372	0.5203	0.4561	0.5122
Baseline + Generic Brown Cluster	0.5219	0.4949	0.5135	0.5068	0.4358	0.4946
Baseline + Generic GloVe Cluster	0.5253	0.4983	0.5405	0.5135	0.4459	0.5047
Baseline + Drug-related GloVe Cluster	0.5219	0.4750	0.5304	0.4797	0.4257	0.4866
All	0.5320	**0.5118**	**0.5642**	**0.5372**	**0.4628**	**0.5216**
All + Spell Check	0.5185	0.4983	0.5405	0.5236	0.4493	0.5061

We observe a significant performance differences between different cluster features. Overall, pre-trained drug-related cluster contributes the highest performance improvement by 3.1%; while for Fold 3, generic GloVe cluster achieves the highest accuracy. This suggests both drug-related and generic clusters are helpful in capturing unseen tokens.

We also investigate the effectiveness of spell correction (the last row of Table 1). However, the performance drops after adding spell check. This probably because we do not include any Twitter specific dictionaries when building the index.

4.2. *Evaluation Results*

Table 2 presents the official results of our three submissions. Run 1 uses All feature set (Section 4.1); Run 2 adds spell correction; Run 3 uses All feature set but excludes the 585 tweets that do not contain any annotated ADR or Indication mentions from the training data. We also include the highest submission from the other participating system for comparison.

Table 2. Comparison of our system (DLIR) with the other participating system on evaluation data.

System	P	R	F1
DLIR_Run 1	0.805	0.482	0.603
DLIR_Run 2	**0.806**	0.485	0.606
DLIR_Run 3	0.760	**0.511**	**0.611**
NTTMUNSW	0.778	0.414	0.540

As shown from the table, our system (DLIR) significantly outperforms the other participating system. In general, Twitter ADR mention extraction task is more challenging in terms of low Recall. We observe Run 3 achieves the highest F1 measure. This indicates data imbalance does hamper the performance.

5. Conclusion

In this paper, we describe our system used in the Social Mining Shared Task for ADR extraction in Twitter. We focus our efforts on improving Twitter ADR extraction using additional word representations, namely, Brown clusters and K-means clusters, that are generated from large amount of unlabeled generic and drug-related tweets. Our experiments and evaluation results show that cluster features derived from word representations are effective. In future, we hope to investigate the use of Twitter specific spell checking dictionaries.

Acknowledgments

We would like to thank Zhiqiang Toh for the help in generating the word clusters.

References

1. J. Sultana, P. Cutroneo, G. Trifiro, Clinical and economic burden of adverse drug reactions. *J Pharmacol Pharmacother.* **4**, S73-S77 (2013).
2. R. Ginn, P. Pimpalkhute, A. Nikfarjam, et al., Mining Twitter for adverse drug reaction mentions: a corpus and classification benchmark. In *Proceedings of the Fourth Workshop on Building and Evaluating Resources for Health and Biomedical Text Processing (BioTxtM).* (2014).
3. K. O'Connor, A. Nikfarjam, R. Ginn, et al., Pharmacovigilance on Twitter? Mining Tweets for adverse drug reactions. In *American Medical Informatics Association (AMIA) Annual Symposium.* (2014).
4. A. Nikfarjam, A. Sarker, K. O'Conner, R. Ginn, G. Gonzalez, Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. In *American Medical Informatics Association (AMIA).* (2014).
5. R. Leaman, L. Wojtulewicz, R. Sullivan, A. Skariah, J. Yang, G. Gonzalez, Towards internet-age pharmacovigilance: extracting adverse drug reactions from user posts to health-related social networks. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing.* (2010).
6. M. Liu and T. Baldwin, Langid.py: An off-the-shelf language identificaiton tool. In *Proceedings of the ACL 2012 System Demonstratiosn.* (2012).
7. J. Turian, L. Ratinov, Y. Bengio, Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* (2010).
8. C. Colin and H. Guo, The unreasonable effectiveness of word representations for Twitter named entity recognition. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* (2015).
9. C. Manning and D. Klein, Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics.* (2003).
10. G. Miller, WordNet: a lexcical database for English. *Commun ACM.* **38**, 39-41 (1995).
11. W. Chapman, W. Bridewell, P. Hanbury, G. Cooper, B. Buchanan, A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics.* **34**, 301-310 (2001).
12. Z. Toh, B. Chen, J. Su, Improving Twitter named entity recognition using word representations. In *Proceedings of the 2015 Workshop on Noisy User-generated Text of the Association for Computational Linguistics.* (2015)
13. R. Tsai, S Wu, W. Chou, et al., Various criteria in the evaluation of biomedical named entity recognition. *BMC Bioinformatics.* **7**, 92 (2006).