AN ENSEMBLE METHOD FOR BINARY CLASSIFICATION OF ADVERSE DRUG REACTIONS FROM SOCIAL MEDIA

ZHIFEI ZHANG* and JIAN-YUN NIE

Department of Computer Science and Operations Research, University of Montreal, Montreal, QC H3C 3J7, Canada

**E-mail: zhanzhif@iro.umontreal.ca*

nie@iro.umontreal.ca

XUYAO ZHANG

National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P.R. China

E-mail: xyz@nlpr.ia.ac.cn

This paper describes the system we developed for PSB 2016 social media mining shared task on binary classification of adverse drug reactions (ADRs). The task focuses on automatic classification of ADR assertive user posts. We propose a weighted average ensemble of four classifiers: (1) a concept-matching classifier based on ADR lexicon; (2) a maximum entropy (ME) classifier with word-level n-gram features and TFIDF weighting scheme; (3) a ME classifier based on word-level n-grams using naive Bayes (NB) log-count ratios as feature values; and (4) a ME classifier with word embedding features. Our system is ranked 2nd with ADR class F-score of 41.82%.

Keywords: Adverse Drug Reactions; Social Media; Maximum Entropy; N-Grams; Word Embeddings.

1. Introduction

Adverse drug reactions (ADRs) are defined as accidental injuries resulting from correct medical drug use. The research on automatic detection of ADRs is currently receiving significant attention from the medical informatics community.[1] In recent years, user posts on social media have been widely used for ADR detection.[2] The PSB 2016 social media mining shared task[a] aims to detect and extract mentions of ADRs from social media. We focus on one of the three subtasks, i.e., automatic binary classification of ADRs.

This problem is formulated as a binary classification of tweets into the ADR or non-ADR classes. ADR lexicons have been the most widely used resources for ADR detection.[2] Pure lexicon-based methods can achieve fairly high recall but at the cost of low precision, since not all mentions that match with the lexicon are adverse reactions. We can also adopt supervised classification methods for which n-grams are commonly used for developing features. In addition to the TFIDF weighting scheme, we use NB log-count ratios as feature values.[3] N-grams may struggle with rare or unseen tokens, so we generate a low dimensional vector for each text with word embeddings.[4] We finally propose a weighted average ensemble of these four classifiers. Our system is ranked 2nd among all 20 participating systems and is very close to the first one. The code to reproduce our experiments in this paper is publicly available[b].

[a] `http://psb.stanford.edu/workshop/wkshp-smm`

[b] `https://github.com/tjflexic/psb-adr`

2. Methods

2.1. *Concept-matching classifier based on ADR lexicon*

An existing ADR lexicon[c] is directly used for ADR detection. The ADR lexicon includes 13699 concepts from COSTART, SIDER, CHV, and DIEGO_Lab. We denote a tweet by t and the ADR lexicon by L. The output of the concept-matching (CM) classifier is given by,

$$f_{\text{CM}}(t) = \begin{cases} 1 \text{ if } \underset{c \in L}{\exists} c \in t \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

2.2. *Maximum entropy classifier with n-grams*

Maximum entropy classifier has been shown to perform extremely well for this classification problem.[5] Let $\{x_1, \cdots, x_m\}$ be a predefined set of m word-level n-gram features from training tweets T. Let $n_i(t)$ be the number of times the feature x_i occurs in a tweet $t \in T$.

The estimate of $P(c|t)$ in ME with n-grams is $P_{\text{ME}}(c|t) = \frac{1}{Z(t)} \exp(\sum_{i=1}^{m} \lambda_{i,c} F_{i,c}(t))$, where $Z(t)$ is a normalization function, $\lambda_{i,c}$'s are feature-weight parameters, and $F_{i,c}$ is a feature/class function for feature x_i and class c. We propose two weighting schemes to implement the feature/class function.

TFIDF weighting TFIDF weighting scheme sets the feature/class function as,

$$F_{i,c}(t) = \begin{cases} (1 + \log(n_i(t))) \times \log(1 + \frac{|T|+1}{|\{t' \in T | n_i(t') > 0\}|+1}) \text{ if } n_i(t) > 0 \\ 0 \qquad \text{otherwise} \end{cases} \tag{2}$$

The ME classifier with TFIDF weighting gives the output $f_{\text{ME-TFIDF}}(t) = P_{\text{ME}}(\text{ADR}|t)$.

NB log-count ratio NB log-count ratios have been found to be useful feature values for classification.[3] The feature/class function with these ratios is defined as follows,

$$F_{i,c}(t) = \begin{cases} \log(\frac{1+\sum_{t:y(t)=\text{ADR}} n_i(t)}{\sum_{i'=1}^{m} (1+\sum_{t:y(t)=\text{ADR}} n_{i'}(t))} \times \frac{\sum_{i'=1}^{m} (1+\sum_{t:y(t)=\text{non-ADR}} n_{i'}(t))}{1+\sum_{t:y(t)=\text{non-ADR}} n_i(t)}) \text{ if } n_i(t) > 0 \\ 0 \qquad \text{otherwise} \end{cases} \tag{3}$$

where $y(t)$ is the true label for a training tweet t. The ME classifier with NB log-count ratios gives the output $f_{\text{ME-NBLCR}}(t) = P_{\text{ME}}(\text{ADR}|t)$.

2.3. *Maximum entropy classifier with word embeddings*

Word embeddings have been shown to boost the performance in NLP-related tasks.[6] The advantage of using word embeddings is to be able to cope with semantic similarity between words rather than relying on isolated words. Specifically, we use the publicly available 150-dimensional word embeddings[d]. To aggregate the words in each tweet, we simply average the vectors of words (excluding stop words).

[c] `http://diego.asu.edu/downloads/publications/ADRMine/ADR_lexicon.tsv`
[d] `http://diego.asu.edu/Publications/ADRMine.html`

Let W be the vocabulary excluding stop words, and $(w_1, \cdots, w_k)$ be the embedding vector of word $w \in W$. The modified estimate of $P(c|t)$ in ME is $P'_{\text{ME}}(c|t) = \frac{1}{Z(t)} \exp(\sum_{i=1}^{k} \lambda_{i,c} F_{i,c}(t))$, where the feature/class function is,

$$F_{i,c}(t) = \frac{\sum_{w \in t} w_i}{\sum_{w \in t} 1} \tag{4}$$

The ME classifier with word embeddings gives the output $f_{\text{ME-WE}}(t) = P'_{\text{ME}}(\text{ADR}|t)$.

2.4. *Ensemble classifier by weighted average*

Our final classifier is an ensemble of the above four classifiers. More formally, we define the overall probability score as the weighted geometric mean of four classifiers' outputs:

$$f_{\text{EWA}}(t) = \alpha_1 \times f_{\text{CM}}(t) + \alpha_2 \times f_{\text{ME-TFIDF}}(t) + \alpha_3 \times f_{\text{ME-NBLCR}}(t) + (1 - \alpha_1 - \alpha_2 - \alpha_3) \times f_{\text{ME-WE}}(t) \tag{5}$$

where $\alpha_1 \geq 0$, $\alpha_2 \geq 0$, $\alpha_3 \geq 0$ and $\alpha_1 + \alpha_2 + \alpha_3 \leq 1$. We find the best setting of weights via brute force grid search, quantizing the coefficient values in the interval $[0, 1]$ at increments of 0.05. The search is evaluated by maximizing the ADR class F-score on a validation set.

The overall probability score can indicate the final label with a decision threshold θ, i.e.,

$$h(t) = \begin{cases} \text{ADR} & \text{if } f_{\text{EWA}}(t) \geq \theta \\ \text{non-ADR} & \text{otherwise} \end{cases} \tag{6}$$

3. Experiments

3.1. *Experimental settings*

We tokenize each tweet with the TwitterNLP toolkit[7] , and remove the *@user*, *URLs* and non-letters of each tweet. Each tweet is converted to low case and the *#hashtag* of it is changed to *hashtag*. The methods in this paper are implemented with the scikit-learn toolkit.[8] The configuration of our system is listed in Table 1. The official evaluation metric is *ADR F-score*, which is the harmonic mean of the precision and recall for the ADR class.[5]

Table 1. System configuration

Classifier	Feature	Configuration
ME-TFIDF	unigrams, bigrams, trigrams	the top 40000 n-grams ordered by term frequency including stop words
ME-NBLCR	unigrams, bigrams, trigrams	all n-grams including stop words
ME-WE	150-dimensional word embeddings	remove stop words
EWA	—	$\alpha_1 = 0.1, \alpha_2 = 0.2, \alpha_3 = 0.35, \theta = 0.5$

3.2. *Results*

We report the results of different classifiers on the validation set in Table 2. The performance of each individual classifier varies with different metrics. CM has the highest *ADR Recall* but

the lowest *ADR Precision*, because one tweet is recognized as ADR as long as it contains one concept from the ADR lexicon. ME-NBLCR has the highest *ADR Precison* but the lowest *ADR Recall*, because it assigns a fairly high weight to one feature appearing in the ADR class. The ensemble classifier EWA benefits from a trade-off between *ADR Precison* and *ADR Recall*, and reveals a remarkable performance improvement with the highest *ADR F-score*. The ensemble model is very effective to deal with imbalanced data in ADR classification.

Table 2. Performance on the validation set

Classifier	*ADR Precision*	*ADR Recall*	*ADR F-score*
CM	0.1400	0.8981	0.2422
ME-TFIDF	0.3424	0.7094	0.4619
ME-NBLCR	0.8065	0.1887	0.3058
ME-WE	0.2843	0.7434	0.4113
EWA	0.5401	0.5585	0.5492

We evaluate the classification performance and the optimal ensemble weights when the decision threshold θ in Eq. (6) varies from 0 to 1. When $\theta \in [0.5, 0.6]$, the system can have a reasonably good performance, and meanwhile ME-WE has the biggest weight. In general, we observe that embedding-based classifier plays an important role in the ensemble and can further improve the performance of classifiers that use more traditional methods.

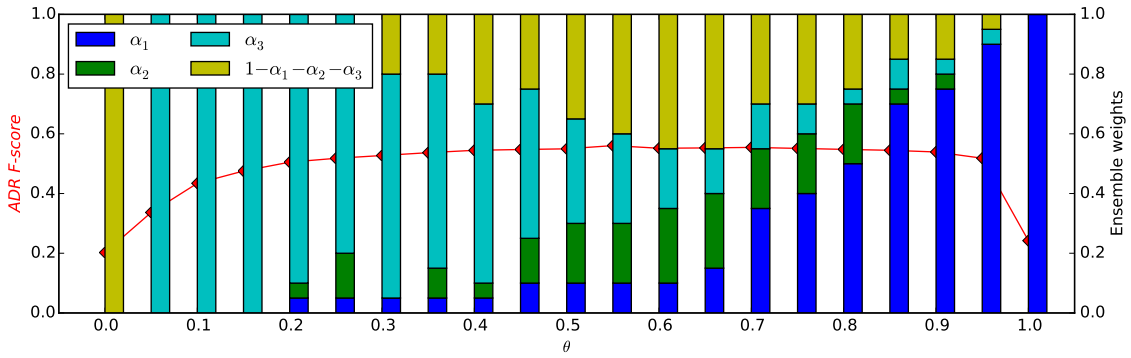

Fig. 1. *ADR F-score* and ensemble weights versus decision threshold on the validation set

4. Conclusions

We propose a simple but effective ensemble system for binary classification of ADRs. We combine four rather complementary classifiers by weighted average aggregation. Each such classifier contributes to the success of the overall system, ranking 2nd on the binary classification of ADRs task of PSB 2016 social media mining shared task workshop. We will further study how word embeddings can be better used and design more useful features for ADR classification.

Acknowledgments

We are really grateful to the organizers and reviewers for this interesting task and their helpful suggestions and comments. This work is partly supported by the National Natural Science Foundation of China (No. 61273304, No. 61363039, No. 61403380), and the Quebec-China Postdoctoral Scholarship (File No. 188040).

References

1. J. Parker, Y. Wei, A. Yates, O. Frieder and N. Goharian, A framework for detecting public health trends with Twitter, in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'13)*, (ACM, New York, 2013).
2. A. Sarker, R. Ginn, A. Nikfarjam, K. O'Connor, K. Smith, S. Jayaraman, T. Upadhaya and G. Gonzalez, Utilizing social media data for pharmacovigilance: A review, *Journal of Biomedical Informatics* **54**, 202 (2015).
3. S. Wang and C. D. Manning, Baselines and bigrams: Simple, good sentiment and topic classification, in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12)*, (ACL, Stroudsburg, 2012).
4. T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, in *Proceedings of the 1st International Conference on Learning Representations*, (Scottsdale, 2013).
5. A. Sarker and G. Gonzalez, Portable automatic text classification for adverse drug reaction detection via multi-corpus training, *Journal of Biomedical Informatics* **53**, 196 (2015).
6. A. Nikfarjam, A. Sarker, K. O'Connor, R. Ginn and G. Gonzalez, Pharmacovigilance from social media: Mining adverse drug reaction mentions using sequence labeling with word embedding cluster features, *Journal of the American Medical Informatics Association* , 1 (2015).
7. K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan and N. A. Smith, Part-of-speech tagging for Twitter: Annotation, features, and experiments, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (ACL, Stroudsburg, 2011).
8. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**, 2825 (2011).