Identifying Personal Experience Tweets of Medication Effects Using Pre-trained RoBERTa Language Model and Its Updating

Minghao Zhu[1,2], **Youzhe Song**[1,2], **Ge Jin**[2], **Keyuan Jiang**[2]
[1]Donghua University, Shanghai, China
[2]Purdue University Northwest, Hammond, Indiana, U.S.A.
minghao.zhu0@gmail.com, isidoresongsisidoresongs@gmail.com, jin9@pnw.edu, kjiang@pnw.edu

Abstract

Post-market surveillance, the practice of monitoring the safe use of pharmaceutical drugs is an important part of pharmacovigilance. Being able to collect personal experience related to pharmaceutical product use could help us gain insight into how the human body reacts to different medications. Twitter, a popular social media service, is being considered as an important alternative data source for collecting personal experience information with medications. Identifying personal experience tweets is a challenging classification task in natural language processing. In this study, we utilized three methods based on Facebook's Robustly Optimized BERT Pretraining Approach (RoBERTa) to predict personal experience tweets related to medication use: the first one combines the pre-trained RoBERTa model with a classifier, the second combines the updated pre-trained RoBERTa model using a corpus of unlabeled tweets with a classifier, and the third combines the RoBERTa model that was trained with our unlabeled tweets from scratch with the classifier too. Our results show that all of these approaches outperform the published methods (Word Embedding + LSTM) in classification performance ($p < 0.05$), and updating the pre-trained language model with tweets related to medications could even improve the performance further.

1 Introduction

Personal experience is an important piece of information for health-related surveillance activities. Understanding one's health experience can help gain insight into the status of one's health, changes of one's health condition after the intervention, or the effects related to any medications one took.

Investigating effects related to the use of pharmaceutical products is an important activity of post-market surveillance. First-hand information related to patients' medication use most directly reflects the effects of the medication, beneficially or adversely. In that case, it is necessary to find valuable data sources and construct efficient methods for processing and analyzing this data.

The widespread availability of social media has made it possible for people to share their personal experiences freely online. Twitter is one of the most prevalent social media services, and studies have shown that the data from social media such as Twitter has been applied to many health-related applications. Examples are as follows: drug adverse events (Bian et al. 2012), public health (Paul et al. 2011; Parker et al. 2013), mental health (Coppersmith et al. 2014; Reece et al. 2017), dental pain (Heaivilin et al. 2011), influenza (Lee et al. 2013; Paul et al. 2015; Gesualdo et al. 2013; Aramaki et al. 2011; Byrd et al. 2016; Kagashe et al. 2017), breast cancer (Thackeray et al. 2013), and epidemic outbreak and spread detection (Ji et al. 2012).

Personal experience is about a person's encounters or observations related to his or her life. Personal experience information related to the use of medication is of unique value for post-market surveillance because it is the first-hand information that reflects the health condition changes due to medication usage. Personal Experience Tweets (PETs) related to medication use are a kind of Twitter post expressing one's personal experience and information after the administration of medication. The types of experiences could be undesirable feelings caused by medications' side-effects, or beneficial effects that help improve a

Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis, pages 127–137
November 20, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

medication user's health condition. The collection and understanding of these experiences' information can help promote the safe use of medications and advance our healthcare practices. Here are some examples of PETs related to medication use (the underscored text is for medication effects and the boldfaced for the medication):

"*Slow release **morphine** almost killed me.*"

"*my mother developed bleeding ulcers from **naproxen** and now they switched her to celebrex isnt that just as bad?*"

"*Ill check it out - I have a friend on **Abilify** and hes had some personality changes, IE agitation, hitting stuff, ect.*"

These tweets show that the effects are associated with a person's experience. In contrast, we define a tweet not describing a personal experience as a non-PET. The following are some examples:

*"wish i had some **xanax** to put me to sleep"*

*"**ativan** please help me get some sleep tonight"*

*"i just took a dose of **percocet** with some strippers"*

The above non-PETs, albeit mentioning medications or containing effect expressions, do not reflect the personal experience.

Extracting PETs from various kinds of Twitter posts is challenging because the Twitter data is of abundant noises, and most of the tweets may be irrelevant to personal experience about health conditions. In addition, users usually post tweets with informal and causal styles, without following the rules of grammar and/or spelling. Finally, Twitter users are creative in coining short text to include the needed information within the space limit. These unique characteristics make it more challenging to identify PETs accurately.

2 Related Works

Distinguishing PETs and non-PETs can be treated as a binary classification task. In the conventional machine learning field, algorithms require a set of manually engineered features extracted from the raw text and/or metadata (Jiang et al., 2016; Wijeratne et al., 2017), usually known as feature engineering, and features chosen can significantly impact the classifier's performance. However, extracting/engineering valuable yet optimal features from tweets is difficult due to the limitation of human knowledge and understanding even for the domain experts. Besides, feature engineering extracts features that are typically based on the analysis of statistics regarding information gain usually with little or no direct consideration of the semantics. In other words, conventional machine learning with feature engineering methods may not be optimal for this task.

Efforts of performance improvement have been made in previous research endeavors in the task of predicting personal experience tweets related to medication effects. In one of the earliest efforts, personal pronouns were considered as an important feature (Jiang and Zheng, 2013). Later, Alvaro and colleagues engineered a set of features (Alvaro et al., 2015), and their features include Twitter-specific features, n-grams, punctuation elements, and topics, but the group decided to discard the topic feature due to the significant efforts required and its minimum merit of improving classification performance. A set of 22 engineered features based upon both textual content and metadata of tweets was proposed in constructing a corpus of personal experience tweets (Jiang et al., 2016). Subsequently, Calix and colleagues introduced the concept of deep gramulator to include a textual feature that contains expressions in one class but not in the opposite class, to improve the discriminatory ability of the classification (Calix et al., 2017). Advancement in neural embedding, which demonstrated state-of-art results in many classification tasks on textual data, motivated the development of a new approach of combining word embedding (word2vec) and a recurrent neural network which demonstrated a significant improvement of classification performance ($p < 0.05$) (Jiang et al., 2018).

Thanks to the development of word embedding techniques and the long short term memory (LSTM) neural network, Jiang et al. (2019) assessed a set of different word embedding techniques: GloVe (Pennington et al. 2014), fastText (Bojanowski et al. 2016) and word2vec (Mikolov et al. 2013) to build vector space models (VSM) to represent the semantics of tweets by learning from a corpus of 22 million unlabeled tweets. The vector representations of tweets were fed into an LSTM neural network for classification. All of these methods achieved better performance in classification measures than the previous methods with 22 human-engineered features using conventional classification algorithms (Jiang et al. 2016).

Unlike the word embedding + LSTM method, which need to learn the VSM first and then train the LSTM network from scratch for classification, Google introduced a fine-tuning based approach by proposing the Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al. 2018), which achieved record-breaking results in 18 downstream NLP tasks. Besides, Google's new method relies on contextual information rather than term co-occurrences. After that, Facebook made some optimization based on BERT and released a Robustly Optimized BERT Pretraining Approach (RoBERTa) model (Liu et al. 2019) which generated even better performance than BERT in downstream tasks. One important and useful aspect of both approaches is that the pre-trained models can be updated with new data, without the need to generate a new model from scratch with the added data, which generally requires a significant amount of computation resources.

In this study, we set the performance of the word embedding + LSTM neural network method as the baseline and investigated the performance improvements of PETs prediction with the pre-trained RoBERTa language model. We also studied a procedure of updating the pre-trained RoBERTa language model and training the RoBERTa from scratch with the medication-related tweets and analyzing the impact on the performance change.

3 Method

In this work, we introduced three ways to identify personal experience tweets about medication effects by using RoBERTa language model: (1) Pretrained RoBERTa - adding a classifier to the standard pre-trained RoBERTa model and fine-tuning the model for classification; (2) Updated RoBERTa - updating the pre-trained RoBERTa language model with our dataset first, then adding a classifier to RoBERTa and fine-tuning the model for classification; and (3) Twitter RoBERTa - training the RoBERTa language model with our corpus of unannotated tweets from scratch, then adding a classifier for classification. Finally, 10-fold cross-validation was performed to gather the performance data, and statistical analysis was performed to determine if the differences in performance among different methods were due to the chance.

The pipelines of data processing and analysis is illustrated in Figure 1. Our process started with gathering Twitter data and performing text encoding after preprocessing. Afterwards, the encoded texts were used with the RoBERTa model and the classifier for our methods. The left pipeline is for the Pretrained RoBERTa approach, the middle one for Updated RoBERTa, and the right one for Twitter RoBERTa.

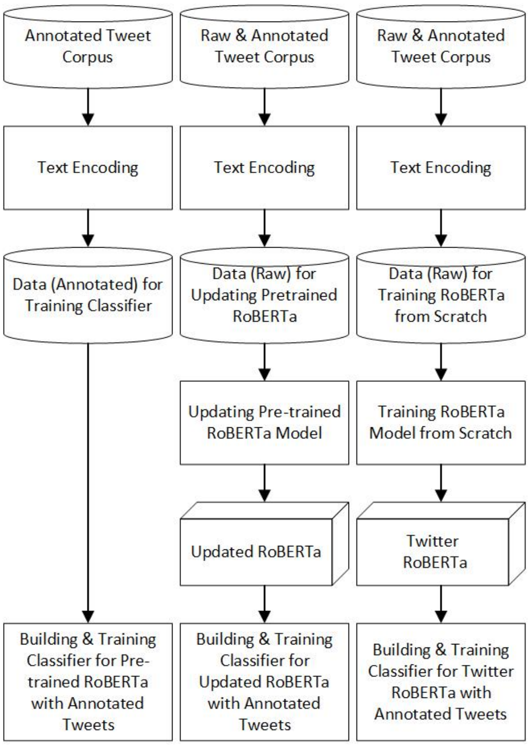
Annotated Tweet Corpus
Raw & Annotated Tweet Corpus
Raw & Annotated Tweet Corpus
Text Encoding
Text Encoding
Text Encoding
Data (Annotated) for Training Classifier
Data (Raw) for Updating Pretrained RoBERTa
Data (Raw) for Training RoBERTa from Scratch
Updating Pre-trained RoBERTa Model
Training RoBERTa Model from Scratch
Updated RoBERTa
Twitter RoBERTa
Building & Training Classifier for Pre-trained RoBERTa with Annotated Tweets
Building & Training Classifier for Updated RoBERTa with Annotated Tweets
Building & Training Classifier for Twitter RoBERTa with Annotated Tweets

Figure 1. The pipelines of data processing.

3.1 Text Encoding

Byte-Pair Encoding (BPE) (Sennrich et al. 2015) and Attention Mask were applied to encode raw text. BPE is a sub-word level encoding method that uses bytes as the base sub-word units. In the process of tokenization, tokens like acronyms, abbreviations or spelling mistakes which are not in the vocabulary are split into known sub-word tokens, Compared to the word-level encoding method, it is flexible enough for tokenized words with special forms and adaptable for most of English documents, and also it could efficiently avoid most of the unknown tokens in the input text. A sub-word vocabulary with 50K unique tokens was built before pre-training, which was tested with our dataset to ensure that our data could be completely covered by this vocabulary and tweet text was tokenized properly without leaving any unknown tokens. In that case, we reused this sub-word vocabulary to encode our data and each of the tweets was converted into a sequence of indices of tokens in the vocabulary.

After encoding, each tweet started with a special `<s>` token and ended with `</s>`. To achieve the fixed length of a sequence, we set the max token length to 64, and a special `<pad>` token was introduced to pad sequences to the max length. We ensured that this value of max token length could fit almost all of the tweets: only 0.003% of them were longer than 64 tokens. Also, an Attention Mask was applied to all of the input data to avoid performing attention on padding tokens. For each sentence, 0 is for padding tokens that should be masked, and 1 is for others that are not masked. Figure 2 shows an example of text encoding.

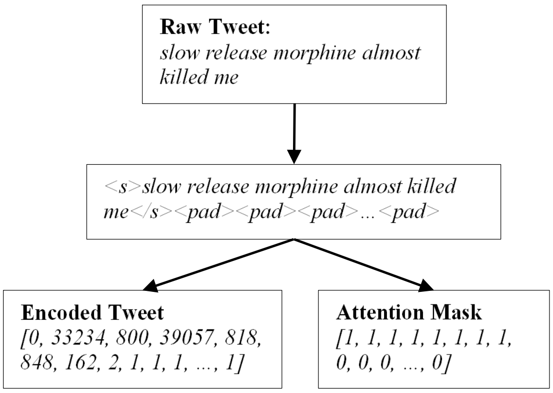

Figure 2. Example of text encoding

3.2 Pre-training

Pre-training the language model in a large corpus could help the model learn a series of general common properties of the language, and it is expected to be used in some of the downstream target tasks with a small dataset where it could perform better. The pre-training model we used is based upon the model of RoBERTa, whose structure is based on Google's BERT model, with 12 layers, 768 hidden neurons, 12 self-attention heads and a total of 110M parameters. The RoBERTa model was released by Facebook AI (Liu et al. 2019), pre-trained with masked language model (MLM) task: 15% of tokens were randomly and dynamically selected for replacement; 80% of them were replaced by a special token `<mask>`; 10% were kept unchanged; the rest of 10% of the tokens were replaced by a random token in vocabulary. The pre-training procedure was performed on a total of over 160GB uncompressed texts for 500K steps with an 8K batch size.

3.3 Language Model (LM) Updating

Although the pre-trained model extracts the general features of linguistic expression in a large corpus, the dataset of our task could be in a different distribution. To make the pre-trained model adapt to our task, we updated the pre-trained RoBERTa model with our corpus of 10M unlabeled tweets before training the classifier. In this updating procedure, we implemented the same masking strategy as that of the masked LM task in the pre-training procedure, described previously, with a set of newly designated hyperparameters *(training steps: 53K/106K/160K batch size: 64, optimizer: Adam, learning rate* 2×10^{-5}*)*

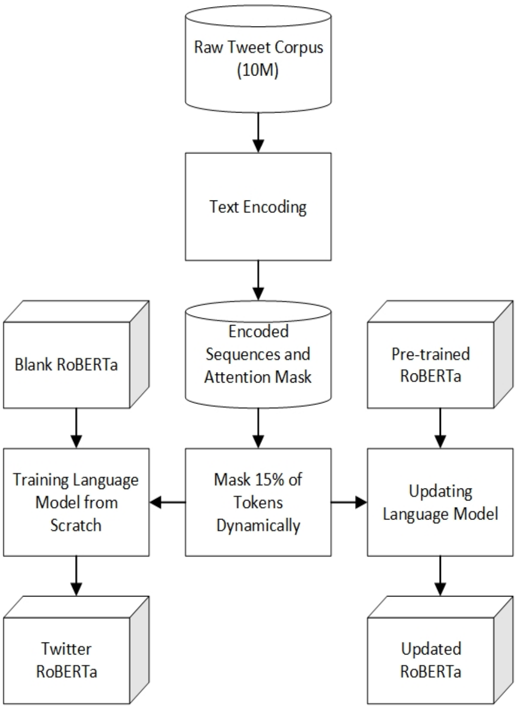

Figure 3. Setup of Updated RoBERTa and Twitter RoBERTa

3.4 Training RoBERTa from Scratch

Another way to let the model learn the property and distribution of a new language environment is to train a new model from scratch with the new dataset. As for our task, it is also a selectable approach. To determine whether training the RoBERTa model with our corpus of tweets could perform better than Facebook's pre-trained one and to use the updating approach in predicting personal experience tweets, a new Twitter RoBERTa model was constructed with the same corpus of tweets as the updating procedure use. Due to the hardware difference between Facebook's and ours, a set of different hyperparameters were used to train it from scratch. *(training steps: 53K/106K/160K, optimizer: Adam, learning rate: 5×10^{-5}, batch size: 64)* Figure 3 illustrates the overview of the procedure of LM updating and training the Twitter RoBERTa from scratch.

3.5 Classifier Fine-tuning

A classifier with a simple feedforward neural network was constructed by following RoBERTa's original design, which is adapted for RoBERTa's base concepts and structure. This is also officially recommended to use for the most of downstream classification tasks by Facebook AI. The classifier is made up of one hidden layer containing 768 units and a tanh activation function followed by a sigmoid output. Between the RoBERTa model and its classifier, the first dimension of RoBERTa's output tensor (also annotated as the beginning of sentence token <s>) was extracted and treated as the input of the classifier. A dropout with a rate of 0.1 was added before the hidden layer to prevent overfitting. We utilized this classifier structure for all of our three methods and fine-tuned the whole model with officially recommended hyperparameters *(epochs: 2, batch size: 32, optimizer: Adam, learning rate: 1×10^{-5})*.

3.6 Baselines

Jiang and colleagues (2018; 2019) investigated and published a set of outstanding methods based on Word Embedding algorithms and the LSTM neural network, which outperformed those using human-engineered features with conventional classification models. Using a large corpus of unlabeled tweets, their approach generated a vector space model (VSM) to encode the words and trained and tested an LSTM-based classifier with a smaller set of annotated tweets. In our approach, we built the same (baseline) models by following the published structures and procedures: a VSM built by word2vec, GloVe and fastText algorithms with 128 dimensions and an LSTM layer with 128 hidden units and L2 regularizer followed by a fully connected layer with the sigmoid output. The models were trained by an Adam optimizer with a learning rate of 2×10^{-4} and a batch size of 32 for 5 epochs.

3.7 Data

Two corpora of Twitter data were used in our work. A total of 22 million raw tweets were collected using Twitter Streaming APIs from August 25, 2015, to December 7, 2016, and another set of 52 million raw tweets was collected from 2006 to 2017 using a home-made crawler based upon the permission policy specified in Twitter's robots.txt file. Both sets were gathered by searching tweets with the keywords of a set of brand and generic medication names. These two corpora were merged and filtered. After dropping duplicates and eliminating non-English twitters, a corpus of 10 million tweets was collected. To study the changes in classification performance, the same corpus of 12,331 annotated tweets, published on Github by (Jiang, et al., 2018), was utilized.

For this task, the corpus of 10 million cleaned tweets were selected for training the Twitter RoBERTa model from scratch as well as updating the LM – note that the both LM updating and training from scratch procedures did not use any labels of the annotation and the annotated 12K tweets were excluded from the 10 million tweets. Interestingly, the baseline methods used the same 10 million raw tweets to build vector space models of neural embedding. Likewise, the baseline classifiers were also trained and tested with 12,331 labeled tweets. Table 1 lists the composition of annotated tweets.

	PETs	Non-PETs	Total
Tweet Count	2,962	9,369	12,331

Table 1. Composition of annotated tweets.

3.8 Statistical Analysis

To determine if any differences in the results among different methods could be due to chance, we conducted statistical analyses on the results between our methods and baseline methods. In our hypothesis testing, the null hypothesis was that the difference between a pair of method does not exist (null hypothesis) while the data remain the same. To do so, we partitioned data into the same subsets for all the methods in cross-validation – that is, each fold has the same set of tweets for different methods. This treatment facilitated us to use the paired t-test on the performance measures of each pair of the method. We set the *p*-value threshold to 0.05, meaning that any *p*-value less than 0.05 ($p <$ 0.05) indicates that the difference does exist and it is not due to chance.

Method	Accuracy	Precision (PET)	Recall (PET)	F1 (PET)	AUC/ROC (PET)
Updated RoBERTa(160K)[1]	0.873	0.732	0.760	0.745	0.933
Updated RoBERTa(106K)[1]	**0.879**	**0.751**	0.746	0.748	**0.934**
Updated RoBERTa(53K)[1]	0.877	0.734	**0.775**	**0.754**	0.932
RoBERTa Original[2]	0.866	0.712	0.759	0.735	0.925
Twitter RoBERTa(160K)[3]	0.859	0.690	0.762	0.724	0.921
Twitter RoBERTa(106K)[3]	0.859	0.706	0.730	0.718	0.917
Twitter RoBERTa(53K)[3]	0.855	0.699	0.709	0.704	0.911
word2vec-LSTM	0.844	0.693	0.661	0.677	0.898
GloVe-LSTM	0.839	0.683	0.651	0.667	0.892
fastText-LSTM	0.842	0.681	0.663	0.672	0.891

1 Updated RoBERTa in 160k, 106k, 53k steps

2 Facebook's pre-trained RoBERTa

3 Twitter RoBERTa in 160k, 106k, 53k steps

Table 2. Classification performance. The last 3 rows are for baseline methods.

4 Results

To compare the performance differences between our methods and baseline methods, 10-fold cross-validation was conducted for each method and the mean value of each classification measure was collected. Table 2 shows the measures of the classification performance between our methods and baselines' (the highest values are in boldface).

Table 3 (in appendix) lists the statistical analysis results of each performance measure in cross-validation between our methods and baseline methods.

5 Discussions

According to the results in Table 2, we can see that compared to baseline methods, the approaches of RoBERTa model with or without updating achieved better performance in all measures, and the Twitter RoBERTa model trained with our data also performed better except in precision, and such differences were confirmed to exist statistically by the *p*-values in Table 3 ($p < 0.05$). In general, we can consider that the RoBERTa models performed better than Word Embedding + LSTM method in this task.

A noticeable improvement between pre-trained and updated RoBERTa models and baseline methods is the precision and recall, whereas the precision of Twitter RoBERTa model remained relatively unchanged at the same time. The recall is the sensitivity of how many true instances are predicted correctly and precision rates how many positive predictions are correct. A higher recall could help the model discover more potential positive instances and higher precision means more true positives (TP) and less false positives (FP) in the prediction. In other words, RoBERTa models can improve the sensitivity and identify PETs more precisely, resulting in more true positives in the predicted PET class.

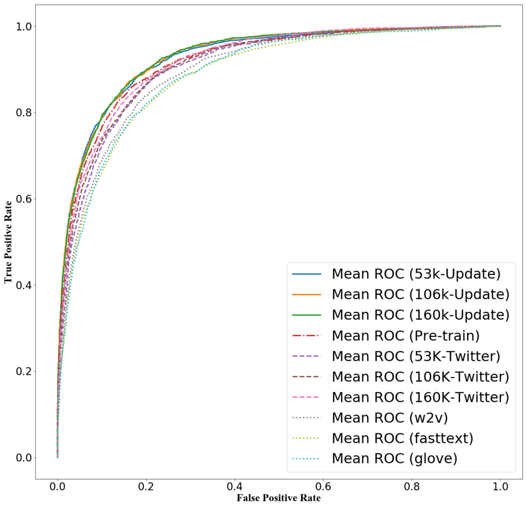

Figure 4. The ROC curves of our methods.

Another remarkable measure could be the ROC/AUC score, which was also improved significantly as shown by the curves in Figure 4. ROC (Receiver Operating Characteristic) is a curve plotting true positive rate (TPR, or sensitivity) in the y-axis and false positive rate (FPR, or 1-specificity) in the x-axis, and is commonly used to show how well the model can distinguish two different objects. The area under the curve (AUC) of ROC is used to quantify the score of ROC. The results in Table 3 show that the lowest *p*-value between our methods and baseline methods is ROC, which may imply that ROC was improved most significantly among all performance measures. That is to say, our methods can be good choices with improved ROCs in this task and they are much more robust in distinguishing PETs and non-PETs.

Our methods also achieved a modest improvement in accuracy, but it could not be interpreted as that better accuracy leads to better performance. Because our dataset is imbalanced (PETs: non-PETs = 1: 3.16, as shown in Table 1) and accuracy is based upon the prediction of both positive and negative classes, higher accuracy could be attributed to the imbalance. Thus, accuracy is not an important measure that should be of concern.

The results also show that performing LM updating before classifier fine-tuning could yield more improvement in accuracy, precision, F1, and AUC. Nevertheless, the *p*-values indicate that they are not significant if updating the LM for more steps. But as for the Twitter RoBERTa model, which was trained from scratch, the steps of training affected performances in some measures which were supported by our statistical analysis. This outcome suggests that a larger number of steps are needed for performance improvement when training from scratch, and small steps are enough for LM updating to achieve better performance than the original RoBERTa model.

The possible reason for the improvement of these RoBERTa-based methods over baseline approaches could be attributed to the level of features. As is known, the features extracted by VSM such as word2vec, which is based upon word-level and co-occurrence. But RoBERTa, which extracts contextual-level features, maybe more powerful in processing tweet-like text which is poisoned by misspelling and incorrect grammars. The possible explanation for the performance difference between Updated RoBERTa and Twitter RoBERTa can be the slow learning process. The updating process is based on the pre-trained RoBERTa model, which is already pre-trained with a very large dataset by Facebook. It may be easier to adapt itself to our dataset, and the larger number of updating steps did less to help improve performance. But for Twitter RoBERTa, since it was trained from scratch and only 15% of tokens were randomly masked, the model could only learn a small part of sentences for each step. Therefore, it may take more time to learn the data distribution, and the larger number of training steps is recommended.

6 Conclusion

In this study, we investigated different ways to use Facebook's RoBERTa model to improve performance in predicting personal experience tweets on medication use. Our results demonstrated that using the fine-tuning method on the pre-trained RoBERTa model achieved better classification performance than previous Word Embedding + LSTM methods, and the original pre-trained RoBERTa could perform better than training a new RoBERTa model from scratch. More importantly, updating the pre-trained RoBERTa language model with our data could yield better performance. The 10-fold cross-validation was used to test statistically the performance differences between our approaches and baseline methods. The results confirmed that the improvement does exist with statistical significance ($p < 0.05$). This suggests the pre-trained RoBERTa model and LM updating method are better choices for this task and significantly boost the capability to identify personal experience tweets. It is conceivable that our method could apply to other classification tasks using Twitter data related to health issues.

7 Acknowledgement

Authors wish to thank College of Technology at Purdue University Northwest for providing funding to support this work.

8 References

Alvaro, N., Conway, M., Doan, S., Lofi, C., Overington, J. and Collier, N., 2015. Crowdsourcing Twitter annotations to identify first-hand experiences of prescription drug use. Journal of biomedical informatics, 58, pp.280-287.

Aramaki, E., Maskawa, S. and Morita, M., 2011, July. Twitter catches the flu: detecting influenza epidemics using Twitter. In Proceedings of the conference on empirical methods in natural language processing (pp. 1568-1576). Association for Computational Linguistics.

Bian, J., Topaloglu, U. and Yu, F., 2012, October. Towards large-scale twitter mining for drug-related adverse events. In Proceedings of the 2012 international workshop on Smart health and wellbeing (pp. 25-32). ACM.

Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T., 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, pp.135-146.

Byrd, K., Mansurov, A. and Baysal, O., 2016, May. Mining Twitter data for influenza detection and surveillance. In Proceedings of the International Workshop on Software Engineering in Healthcare Systems (pp. 43-49). ACM.

Calix, R.A., Gupta, R., Gupta, M. and Jiang, K., 2017, November. Deep gramulator: Improving precision in the classification of personal health-experience tweets with deep learning. In 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 1154-1159). IEEE.

Coppersmith, G., Dredze, M. and Harman, C., 2014, June. Quantifying mental health signals in Twitter. In Proceedings of the workshop on computational linguistics and clinical psychology: From linguistic signal to clinical reality (pp. 51-60).

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Gesualdo, F., Stilo, G., Gonfiantini, M.V., Pandolfi, E., Ve-lardi, P. and Tozzi, A.E., 2013. Influenza-like illness surveil-lance on Twitter through automated learning of naïve language. PLoS One, 8(12), p.e82489.

Heaivilin, N., Gerbert, B., Page, J.E. and Gibbs, J.L., 2011. Public health surveillance of dental pain via Twitter. Journal of dental research, 90(9), pp.1047-1051.

Ji, X., Chun, S.A. and Geller, J., 2012, April. Epidemic outbreak and spread detection system based on twitter data. In International Conference on Health Information Science (pp. 152-163). Springer, Berlin, Heidelberg.

Jiang, K., Calix, R. and Gupta, M., 2016, August. Construction of a personal experience tweet corpus for health surveillance. In Proceedings of the 15th workshop on biomedical natural language processing (pp. 128-135).

Jiang, K., Feng, S., Calix, R.A. and Bernard, G.R., 2019, January. Assessment of word embedding techniques for identification of personal experience tweets pertaining to medication uses. In International Workshop on Health Intelligence (pp. 45-55). Springer, Cham.

Jiang, K., Feng, S., Song, Q., Calix, R.A., Gupta, M. and Bernard, G.R., 2018. Identifying tweets of personal health experience through word embedding and LSTM neural network. BMC bioinformatics, 19(8), p.210.

Jiang, K. and Zheng, Y., 2013, December. Mining twitter data for potential drug effects. In International conference on advanced data mining and applications (pp. 434-443). Springer, Berlin, Heidelberg.

Kagashe, I., Yan, Z. and Suheryani, I., 2017. Enhancing seasonal influenza surveillance: topic analysis of widely used medicinal drugs using Twitter data. Journal of medical Internet research, 19(9), p.e315.

Lee, K., Agrawal, A. and Choudhary, A., 2013, August. Real-time disease surveillance using twitter data: demonstration on flu and cancer. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1474-1477). ACM.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Parker, J., Wei, Y., Yates, A., Frieder, O. and Goharian, N., 2013, August. 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 (pp. 556-563). ACM.

Paul, M.J. and Dredze, M., 2011, July. You are what you tweet: Analyzing twitter for public health. In Fifth International AAAI Conference on Weblogs and Social Media.

Paul, M.J., Dredze, M., Broniatowski, D.A. and Generous, N., 2015, April. Worldwide influenza surveillance through twitter. In Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence.

Pennington, J., Socher, R. and Manning, C., 2014, October. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Reece, A.G., Reagan, A.J., Lix, K.L., Dodds, P.S., Danforth, C.M. and Langer, E.J., 2017. Forecasting the onset and course of mental illness with Twitter data. Scientific reports, 7(1), p.13006.

Sennrich, R., Haddow, B. and Birch, A., 2015. Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909.

Thackeray, R., Burton, S.H., Giraud-Carrier, C., Rollins, S. and Draper, C.R., 2013. Using Twitter for breast cancer prevention: an analysis of breast cancer awareness month. BMC cancer, 13(1), p.508.

Wijeratne, S., Sheth, A., Bhatt, S., Balasuriya, L., Al-Olimat, H. S., Gaur, M., Yazdavar, A. H., Thirunarayan, K.: Feature Engineering for Twitter-based Applications. Feature Engineering for Machine Learning and Data Analytics, 35 (2017).

A Appendices

Method	Measure	RoBERTa	RoBERTa-Updated(53K)	RoBERTa-Updated(106K)	RoBERTa-Updated(160K)	RoBERTa-Twitter(53K)	RoBERTa-Twitter(106K)	RoBERTa-Twitter(160K)
RoBERTa	Acc.		5.106×10^{-2}	**7.908×10^{-4}**	1.036×10^{-1}	**2.153×10^{-2}**	**2.595×10^{-2}**	7.047×10^{-2}
	Prec.		1.793×10^{-1}	**3.950×10^{-3}**	2.075×10^{-1}	2.344×10^{-1}	3.248×10^{-1}	1.071×10^{-1}
	Recall		2.170×10^{-1}	1.651×10^{-1}	4.853×10^{-1}	**4.586×10^{-2}**	1.096×10^{-1}	4.484×10^{-1}
	F1		**8.196×10^{-4}**	**1.247×10^{-3}**	**2.306×10^{-2}**	**5.759×10^{-3}**	**1.576×10^{-2}**	9.804×10^{-2}
	AUC		**1.312×10^{-4}**	**7.787×10^{-5}**	**2.650×10^{-4}**	**2.397×10^{-5}**	**6.578×10^{-4}**	**3.026×10^{-2}**
RoBERTa-Updated(53K)	Acc.	5.106×10^{-2}		3.204×10^{-1}	2.880×10^{-1}	**6.821×10^{-4}**	**4.198×10^{-3}**	**9.187×10^{-4}**
	Prec.	1.793×10^{-1}		1.774×10^{-1}	4.660×10^{-1}	**3.713×10^{-2}**	1.694×10^{-1}	**1.699×10^{-2}**
	Recall	2.170×10^{-1}		**4.363×10^{-2}**	2.656×10^{-1}	**1.410×10^{-2}**	9.040×10^{-2}	3.019×10^{-1}
	F1	**8.196×10^{-4}**		1.094×10^{-1}	**4.049×10^{-2}**	**1.948×10^{-4}**	**1.954×10^{-6}**	**6.061×10^{-5}**
	AUC	**1.312×10^{-4}**		8.087×10^{-2}	5.282×10^{-2}	**9.169×10^{-8}**	**2.980×10^{-7}**	**7.746×10^{-6}**
RoBERTa-Updated(106K)	Acc.	**7.908×10^{-4}**	3.204×10^{-1}		7.762×10^{-2}	**4.148×10^{-5}**	**1.415×10^{-5}**	**6.621×10^{-5}**
	Prec.	**3.950×10^{-3}**	1.774×10^{-1}		1.682×10^{-1}	**5.194×10^{-3}**	**1.010×10^{-2}**	**9.835×10^{-4}**
	Recall	1.651×10^{-1}	**4.363×10^{-2}**		2.882×10^{-1}	1.199×10^{-1}	2.575×10^{-1}	2.289×10^{-1}
	F1	**1.247×10^{-3}**	1.094×10^{-1}		1.449×10^{-1}	**3.382×10^{-4}**	**1.216×10^{-4}**	**2.713×10^{-4}**
	AUC	**7.787×10^{-5}**	8.087×10^{-2}		3.416×10^{-1}	**1.357×10^{-7}**	**5.519×10^{-7}**	**8.113×10^{-6}**
RoBERTa-Updated(160K)	Acc.	1.036×10^{-1}	2.880×10^{-1}	7.762×10^{-2}		**3.212×10^{-3}**	**3.471×10^{-3}**	**4.285×10^{-3}**
	Prec.	2.075×10^{-1}	4.660×10^{-1}	1.682×10^{-1}		6.953×10^{-2}	1.398×10^{-1}	**2.216×10^{-2}**
	Recall	4.853×10^{-1}	2.656×10^{-1}	2.882×10^{-1}		**3.748×10^{-2}**	1.779×10^{-1}	4.702×10^{-1}
	F1	**2.306×10^{-2}**	**4.049×10^{-2}**	1.449×10^{-1}		**1.828×10^{-4}**	**1.204×10^{-3}**	**6.101×10^{-3}**
	AUC	**2.650×10^{-4}**	5.282×10^{-2}	3.416×10^{-1}		**8.982×10^{-9}**	**6.030×10^{-8}**	**7.682×10^{-7}**
RoBERTa-Twitter(53K)	Acc.	**2.153×10^{-2}**	**6.821×10^{-4}**	**4.148×10^{-5}**	**3.212×10^{-3}**		2.124×10^{-1}	1.908×10^{-1}
	Prec.	2.344×10^{-1}	**3.713×10^{-2}**	**5.194×10^{-3}**	6.953×10^{-2}		3.888×10^{-1}	2.640×10^{-1}
	Recall	**4.586×10^{-2}**	**1.410×10^{-2}**	1.199×10^{-1}	**3.748×10^{-2}**		2.817×10^{-1}	**2.878×10^{-2}**
	F1	**5.759×10^{-3}**	**1.948×10^{-4}**	**3.382×10^{-4}**	**1.828×10^{-4}**		1.791×10^{-1}	**2.271×10^{-2}**
	AUC	**2.397×10^{-5}**	**9.169×10^{-8}**	**1.357×10^{-7}**	**8.982×10^{-9}**		**1.873×10^{-3}**	**2.129×10^{-5}**
RoBERTa-Twitter(106K)	Acc.	**2.595×10^{-2}**	**4.198×10^{-3}**	**1.415×10^{-5}**	**3.471×10^{-3}**	2.124×10^{-1}		4.596×10^{-1}
	Prec.	3.248×10^{-1}	1.694×10^{-1}	**1.010×10^{-2}**	1.398×10^{-1}	3.888×10^{-1}		1.801×10^{-1}
	Recall	1.096×10^{-1}	9.040×10^{-2}	2.575×10^{-1}	1.779×10^{-1}	2.817×10^{-1}		8.020×10^{-2}
	F1	**1.576×10^{-2}**	**1.954×10^{-6}**	**1.216×10^{-4}**	**1.204×10^{-3}**	1.791×10^{-1}		**4.668×10^{-2}**
	AUC	**6.578×10^{-4}**	**2.980×10^{-7}**	**5.519×10^{-7}**	**6.030×10^{-8}**	**1.873×10^{-3}**		**1.630×10^{-3}**
RoBERTa-Twitter(160K)	Acc.	7.047×10^{-2}	**9.187×10^{-4}**	**6.621×10^{-5}**	**4.285×10^{-3}**	1.908×10^{-1}	4.596×10^{-1}	
	Prec.	1.071×10^{-1}	**1.699×10^{-2}**	**9.835×10^{-4}**	**2.216×10^{-2}**	2.640×10^{-1}	1.801×10^{-1}	
	Recall	4.484×10^{-1}	3.019×10^{-1}	2.289×10^{-1}	4.702×10^{-1}	**2.878×10^{-2}**	8.020×10^{-2}	
	F1	9.804×10^{-2}	**6.061×10^{-5}**	**2.713×10^{-4}**	**6.101×10^{-3}**	**2.271×10^{-2}**	**4.668×10^{-2}**	
	AUC	**3.026×10^{-2}**	**7.746×10^{-6}**	**8.113×10^{-6}**	**7.682×10^{-7}**	**2.129×10^{-5}**	**1.630×10^{-3}**	

Table 3a. Statistical analysis results (p values) for RoBERTa models. Values in boldface are less than 0.05.

Method	Measure	RoBERTa	RoBERTa-Updated(53K)	RoBERTa-Updated(106K)	RoBERTa-Updated(160K)	RoBERTa-Twitter(53K)	RoBERTa-Twitter(106K)	RoBERTa-Twitter(160K)
Word2Vec-LSTM	Acc.	**1.663×10^{-3}**	**1.879×10^{-5}**	**1.919×10^{-6}**	**1.050×10^{-4}**	**7.198×10^{-3}**	**5.770×10^{-3}**	**4.898×10^{-3}**
	Prec.	1.861×10^{-1}	**4.768×10^{-2}**	**6.659×10^{-3}**	9.901×10^{-2}	4.045×10^{-1}	2.818×10^{-1}	4.422×10^{-1}
	Recall	**2.603×10^{-2}**	**1.062×10^{-2}**	**3.799×10^{-2}**	**2.877×10^{-2}**	1.864×10^{-1}	5.933×10^{-2}	**2.029×10^{-2}**
	F1	**1.145×10^{-2}**	**1.878×10^{-3}**	**2.958×10^{-3}**	**3.891×10^{-3}**	7.551×10^{-2}	**2.553×10^{-2}**	**1.450×10^{-2}**
	AUC	**2.582×10^{-7}**	**8.251×10^{-8}**	**5.581×10^{-8}**	**4.686×10^{-8}**	**1.479×10^{-5}**	**1.864×10^{-5}**	**5.317×10^{-7}**
Glove-LSTM	Acc.	**9.613×10^{-5}**	**8.448×10^{-5}**	**1.213×10^{-5}**	**2.637×10^{-4}**	**1.236×10^{-2}**	**1.137×10^{-3}**	**5.019×10^{-4}**
	Prec.	1.005×10^{-1}	**2.234×10^{-2}**	**3.395×10^{-3}**	**2.055×10^{-2}**	2.617×10^{-1}	1.686×10^{-1}	3.822×10^{-1}
	Recall	**1.442×10^{-2}**	**3.515×10^{-3}**	**1.768×10^{-2}**	**1.818×10^{-3}**	1.008×10^{-1}	**4.343×10^{-2}**	**1.303×10^{-2}**
	F1	**1.155×10^{-4}**	**1.451×10^{-5}**	**2.706×10^{-5}**	**1.673×10^{-5}**	**4.342×10^{-3}**	**1.953×10^{-3}**	**7.256×10^{-4}**
	AUC	**7.183×10^{-9}**	**1.086×10^{-9}**	**3.358×10^{-10}**	**1.338×10^{-10}**	**1.994×10^{-9}**	**1.326×10^{-8}**	**2.239×10^{-9}**
Fasttext-LSTM	Acc.	**9.961×10^{-5}**	**5.171×10^{-5}**	**1.029×10^{-6}**	**2.716×10^{-4}**	**7.583×10^{-3}**	**2.108×10^{-3}**	**7.676×10^{-3}**
	Prec.	**3.035×10^{-2}**	**1.449×10^{-2}**	**1.133×10^{-4}**	**2.920×10^{-2}**	1.864×10^{-1}	1.425×10^{-1}	3.588×10^{-1}
	Recall	**2.448×10^{-5}**	**1.183×10^{-4}**	**4.201×10^{-5}**	**9.241×10^{-4}**	9.009×10^{-2}	**1.946×10^{-2}**	**3.609×10^{-3}**
	F1	**8.453×10^{-8}**	**6.394×10^{-9}**	**3.063×10^{-8}**	**9.138×10^{-8}**	**1.282×10^{-3}**	**1.064×10^{-4}**	**5.055×10^{-6}**
	AUC	**1.011×10^{-8}**	**3.002×10^{-9}**	**1.344×10^{-8}**	**3.410×10^{-9}**	**9.257×10^{-8}**	**2.562×10^{-7}**	**1.066×10^{-8}**

Table 3b. Statistical analysis results (*p* values) for baselines. Values in boldface are less than 0.05.