Identifying Medication Abuse and Adverse Effects from Tweets: University of Michigan at #SMM4H 2020

V.G.Vinod Vydiswaran,[1,2†] **Deahan Yu,**[2] **Xinyan Zhao,**[2] **Ermioni Carr,**[2]
Jonathan Martindale,[2] **Jingcheng Xiao,**[3] **Noha Ghannam,**[2] **Matteo Althoen,**[2]
Alexis Castellanos,[4] **Neel Patel,**[5] **Daniel Vasquez**[2]
[1]Department of Learning Health Sciences, University of Michigan Medical School
[2]School of Information, University of Michigan; [3]College of Pharmacy, University of Michigan
[4]College of Engineering & Computer Science, University of Michigan, Dearborn
[5]Internal Medicine Department, Garnet Health Medical Center
[†] Corresponding author: vgvinodv@umich.edu

Abstract

The team from the University of Michigan participated in three tasks in the Social Media Mining for Health Applications (#SMM4H) 2020 shared tasks – on detecting mentions of adverse effects (Task 2), extracting and normalizing them (Task 3), and detecting mentions of medication abuse (Task 4). Our approaches relied on a combination of traditional machine learning and deep learning models. On Tasks 2 and 4, our submitted runs performed at or above the task average.

1 Introduction

The Social Media Mining for Health Applications (#SMM4H) Shared Task 2020 provided a unique hands-on experience for graduate students and researchers to apply concepts in text cleaning, natural language processing, and health informatics on standardized research tasks. The tasks are motivated by challenges in using social media data for health research, including informal, colloquial expressions and misspelling of clinical concepts, data sparsity, noise, and ambiguity (Klein et al., 2020).

This year, a team of students from the University of Michigan participated in three tasks – on classification of tweets that report adverse effects (Task 2 English), extraction and normalization of adverse effects (Task 3), and characterization of chatter related to prescription medication abuse (Task 4). We submitted three runs for each of these tasks. This paper describes our approach in developing and training the components for our participation, along with the results over validation and test data sets.

2 Task 2: Classifying tweets that report adverse effects

This binary classification task aimed at distinguishing tweets that report a medication adverse effect (AE) (labeled "positive"), from those that do not (labeled "negative"), taking into account subtle linguistic variations between adverse effects and "indication" or the reason for using the medication (Weissenbacher et al., 2019). The training data (n=25,672) was split into train (n=20,544; 80%) and validation (n=5,134; 20%) sets. 2,374 (9.2%) tweets in the training data were "positive". The test data had 4,759 tweets.

We participated in the previous iteration of this task in #SMM4H 2019, and proposed two support vector machine (SVM) models and a bidirectional long short-term memory (LSTM) model (Vydiswaran et al., 2019). We followed up on our participation during the post evaluation phase to further develop our approaches and proposed a neural network (NN) model called the collocated LSTM with attentive pooling and aggregated representation (CLAPA) (Zhao et al., 2019), that performed better than the three official submissions. Based on those insights, this year we proposed three NN models, viz. a multi-stage convolutional neural network model, the CLAPA model, and a BERT-augmented CLAPA model.

Data cleaning and preprocessing: We removed all duplicate tweets between the training and validation data sets, removed URLs and non-alphanumeric characters, and normalized the Twitter user mentions by replacing them with a token "username". We also applied the `emoji` library[1] in Python to substitute the emoticons with synonymous descriptive text. The tweets were tokenized and padded to normalize them to the same length.

[1]The `emoji` library. Last accessed 2020 Jul 7. https://pypi.org/project/emoji/

Proceedings of the 5th Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task, pages 90–94
Barcelona, Spain (Online), December 12, 2020.

Run 1 – Convolutional Neural Network (CNN): The CNN model consisted of an input representation using Twitter Word2Vec embeddings (Godin, 2019) fed into a convolutional layer of 128 filters and 5 kernels, followed by a global max pooling layer to reduce input dimensionality, and a dropout layer to reduce overfitting. A sigmoid activation function is used to output the final label. The model parameters were tuned over the validation set, and used to train the final model on train and validation sets.

Run 2 – CLAPA: We retrained a model we proposed last year, called CLAPA – a collocated LSTM model with attentive pooling and aggregated representation (Zhao et al., 2019). This model takes a concatenation of word embbedings and collocated embedding into an LSTM layer, and passes the final hidden states to an attentive pooling layer. These pooled states are aggregated with medical collocation information and fed into a fully connected layer for final prediction label.

Based on the model described by (Zhao et al., 2019), medical concepts were constructed over the training set and expanded with a list of drugs from MedlinePlus[2]. A collocation graph was constructed with nodes as unique words in the dataset, and undirected edges from each medical concept to its 15 closest word neighbors. FastText pretrained word embedding (Joulin et al., 2017) was used as the input representation. The model consists of one layer of LSTM network with the hidden size as 300 and three multi-head attention layers.

Run 3 – BERT-augmented CLAPA: As our third run, we developed an extension to CLAPA with Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019). We extended CLAPA to utilize BERT's logits because BERT encodes a powerful, yet functionally different, representation for a given input. Specifically, the logits generated by BERT focus on representations from the full context of a tweet while the logits generated by CLAPA focus heavily on external medical domain information. While both CLAPA and BERT are independently trained on the dataset, in this BERT-augmented CLAPA model, we configure BERT to pass its logits to the CLAPA to allow the CLAPA model to make prediction based on the concatenation of the two logits.

3 Task 3: Extracting and normalizing adverse effects mentioned in tweets

This task involves extracting the span of text containing an AE of a medication from tweets that report an AE, and then mapping the extracted AE to a standard concept ID in the Medical Dictionary for Regulatory Activities (MedDRA) vocabulary (Mozzicato, 2009). The training data includes tweets that report an AE (annotated as "positive") and those that do not (annotated as "negative"). For each "positive" tweet, the training data contains the span of text containing the AE, the character offsets of that span of text, and the MedDRA ID of the AE. There were 2,376 tweets in the training set, of which 1,212 (51%) were "positive". The test set had 1,000 tweets.

Data cleaning and preprocessing: We removed all usernames, URLs, and non-alphanumeric characters in the tweets, and expanded all the contractions to normalize the text. To avoid downstream challenges of the preprocessing steps changing the character offsets of subsequent tokens, we created a character offset map for tokens in the raw tweet. Finally, we changed the tweet text to lowercase, and tokenized it using NLTK's `twitter_tokenizer` package.

We formulated the adverse effect extraction task as a sequence classification problem. We prepared the tokenized tweet text for this task by tagging each token in the training set with one of B-AE, I-AE, or O tags, to indicate the beginning, inside, and outside an AE span, respectively. Regular expression-based matching was used to identify the annotated spans in the training set.

Model development: All three runs that we submitted for Task 3 were based on a standard bidirectional long short-term memory (Bi-LSTM) model, followed by a conditional random field (CRF) model. However, the three models differed on the word embedding used to represent the tweet tokens, in model hyperparameters, and in the MedDRA Concept Unique Identifier (CUI) lists for the normalization task.

[2]MedlinePlus [Internet]. Bethesda (MD): National Library of Medicine (US); [updated 2020 Jul 6]. Drugs, Herbs and Supplements; [updated 2015 Apr 28; cited 2020 Jun 29]. Available from: https://medlineplus.gov/druginformation.html

Run 1 – Bi-LSTM CRF + MedDRA: For Run 1, we concatenated the representation from two word embbedings – viz. GloVe (Pennington et al., 2014) and EXT (Komninos and Manandhar, 2016) – and fed into a bi-LSTM model, followed by a CRF layer for this sequence classification task. The size of the hidden layer in the LSTM model was set as 75, the batch size was set as 64, and learning rate was 0.005.

To normalize the extracted text to a MedDRA term code, we applied QuickUMLS (Soldaini and Goharian, 2016) on the extracted text. QuickUMLS performs a fuzzy string match to find potential candidate CUI matches. When two or more CUIs were identified as candidates, the UMLS-MedDRA[3] list was used to choose the first candidate match as our normalized MedDRA CUI output.

Run 2 – Bi-LSTM CRF + MedDRA with different hyperparameters: The model architecture is similar to the one used in Run 1, except that we used only the EXT word embedding as the input representation. The size of the hidden layer in the LSTM model was set as 55, the batch was set as 32, and learning rate was 0.001. The approach used for normalization was identical to the one used in Run 1.

Run 3 – Bi-LSTM CRF with a modified version of MedDRA: For Run 3, the extraction model was identical to that used in Run 1, but the normalization method was different. Instead of the UMLS-MedDRA list, we used a modified version of the dictionary. This modified version was created by first identifying examples of the most commonly missed phrases in the training set, and then manually adding them based on their concept identifiers in the UMLS Metathesaurus (Bodenreider, 2004).

4 Task 4: Characterizing prescription medication abuse in tweets

This multi-class classification task involves distinguishing, among tweets that mention at least one prescription opioid, benzodiazepine, atypical anti-psychotic, central nervous system stimulant, or GABA analogue, tweets that report potential abuse/misuse (annotated as "A"), non-abuse/-misuse consumption (annotated as "C"), merely mention medications (annotated as "M"), or are unrelated (annotated as "U") (O'Connor et al., 2020). The training and test sets consist of 13,172 and 3,271 tweets, respectively.

The evaluation metric was set as the F1 score for the potential abuse/misuse class. So, we reformulate the task as a binary classification task of distinguishing potential abuse/misuse tweets (class "A") from non-abuse tweets (union of classes "C", "M", and "U").

Data cleaning and preprocessing: We removed all usernames, URLs, punctuation symbols, and special characters because they did not add substantial value to the classification of the tweet based on our initial analysis of the training set. We also removed all emojis as we observed that they were not as helpful, and often detrimental, to the training of our models, even when they were converted to descriptive tags such as ":smilingface:". We converted tweets to lower case and replaced all drug names, generic and brand name, with a token "drugname" so that models could focus on the context surrounding the mention of a drug, rather than the name of the drug itself.

Runs 1 & 3 – SVM with radial basis function (RBF) and linear kernels: Runs 1 and 3 were based on SVM models trained with an RBF kernel and a linear kernel, respectively. For both models, we set $C = 1.0$ and $\gamma = 0.5$. We also set the class weight parameter as "balanced" to adjust for the uneven distribution of classes in our binary abuse vs. non-abuse formulation. The model parameters were tuned over the validation set, and used to train the final model on train and validation sets. The primary difference between Runs 1 and 3 was that for Run 1, we explicitly balanced the abuse to non-abuse distribution in the train set by sub-sampling the non-abuse class during the training phase.

Run 2 – Sequential neural network: Run 2 was based on a sequential neural network composed of three dense layers utilizing relu activation. The input text was transformed into tf-idf weighted representation and fed into the first dense layer of size 16, followed by a dropout of 0.5. In the second dense layer, the number of training features was reduced to 4, followed by another dropout of 0.5. The output layer was configured to generate a binary output using sigmoid activation. We used a binary cross-entropy loss during optimization, using the Adam optimizer with a learning rate of 0.001 and a decay rate of $1e$-*6*.

[3]MDR (MedDRA) - Synopsis. U.S. National Library of Medicine. Last accessed 2020 Jul 8. https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/MDR/index.html

Task	Run description	Validation F1	Test set F1
Task 2	Run 1: CNN	0.510	0.35
	Run 2: CLAPA	0.603	0.44
	Run 3: BERT-augmented CLAPA	**0.629**	**0.51**
	Averaged best performance (n = 16)	–	*0.46*
Task 3: Extraction	Run 1: Bi-LSTM CRF + MedDRA	0.331	0.463
	Run 2: Bi-LSTM CRF + MedDRA w/ different hyperparameters	**0.350**	0.463
	Run 3: Bi-LSTM CRF + modified MedDRA	0.331	0.463
	Averaged best performance, Relaxed (n = 7)	–	***0.564***
Task 3: Normalization	Run 1: Bi-LSTM CRF + MedDRA	0.112	0.193
	Run 2: Bi-LSTM CRF + MedDRA w/ different hyperparameters	0.115	0.193
	Run 3: Bi-LSTM CRF + modified MedDRA	**0.154**	0.196
	Averaged best performance, Relaxed (n = 7)	–	***0.292***
Task 4	Run 1: SVM with RBF kernel	**0.46**	**0.49**
	Run 2: Sequential neural network	0.41	0.43
	Run 3: SVM with Linear kernel	0.45	0.47
	Averaged best performance (n = 4)	–	***0.49***

Table 1: Summary of the runs submitted by our team, and their F1 scores over validation and test sets. For Task 3, only the results corresponding to the "relaxed" token match are shown.

5 Results

The performance of our submitted runs for all three tasks on validation and test sets are summarized in Table 1. The table also includes the averaged F1 score of the best runs submitted by all participating teams in each task on the test set. In Task 2, the run using the BERT-augmented CLAPA model (Run 3) performed the best, and outperformed the averaged F1 score of best-performing runs submitted by Task 2 participants. We also note that the performance of our submitted runs on the test set were significantly lower than those on the validation set. In Task 3, all three runs we submitted performed similarly, which implies that the variations we tried in the word embedding used to represent the tweet tokens did not affect the overall performance. Run 3 performed slightly better on the normalization task, but was below the average F1 score for the task. Finally, in Task 4, the SVM model with RBF kernel (Run 1) performed the best among our submitted runs, and was at par with the average F1 score of the best-performing runs.

Task	Run description	Precision	Recall	F1
Task 2	Run 3: BERT-augmented CLAPA	**0.48**	0.54	**0.51**
	Averaged best performance (n=16)	*0.42*	*0.59*	*0.46*
Task 3: Extraction	Run 3: Bi-LSTM CRF + modified MedDRA	**0.806**	0.324	0.463
	Averaged best performance, Relaxed (n=7)	*0.607*	*0.557*	*0.564*
Task 3: Normalization	Run 3: Bi-LSTM CRF + modified MedDRA	**0.345**	0.137	0.196
	Averaged best performance, Relaxed (n=7)	*0.312*	*0.290*	*0.292*
Task 4	Run 1: SVM with RBF kernel	0.462	0.513	**0.49**
	Averaged best performance (n=4)	–	–	*0.49*

Table 2: Comparison of precision, recall, and F1 measures of the best runs submitted by our team on the test set, compared to the average of the best-performing runs from all participants in each task.

Table 2 shows additional details of our best-performing runs in terms of precision, recall, and F1 scores on the test set, and the corresponding measures averaged over the best-performing runs from all participants in each task. All of our best-performing runs, including those for Task 3, achieved a higher precision than the average precision of all submitted best-runs, but have lower recall. In future, we plan to further explore ways to improve the recall of our proposed approaches for all three tasks.

6 Conclusion

Our approach for participating in the #SMM4H 2020 Shared Tasks focused on building on our previous efforts in addressing related tasks. We participated in three tasks and experimented with a combination of traditional machine learning and deep learning models. Two of the submitted runs performed at or above the average task performance. Additional experiments are planned to further improve the recall, and thereby the overall performance, of our proposed models for these tasks.

References

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl_1):D267-D270, January.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, June.

Fréderic Godin. 2019. *Improving and Interpreting Neural Networks for Word-Level Prediction Tasks in Natural Language Processing*. Ph.D. thesis, Ghent University, Belgium.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain, April. Association for Computational Linguistics.

Ari Z. Klein, Ilseyar Alimova, Ivan Flores, Arjun Magge, Zulfat Miftahutdinov, Anne-Lyse Minard, Karen O'Connor, Abeed Sarker, Elena Tutubalina, Davy Weissenbacher, and Graciela Gonzalez-Hernandez. 2020. Overview of the fifth Social Media Mining for Health Applications (#SMM4H) Shared Tasks at COLING 2020. In *Proceedings of the Fifth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*.

Alexandros Komninos and Suresh Manandhar. 2016. Dependency based embeddings for sentence classification tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1490–1500, San Diego, California, June. Association for Computational Linguistics.

Patricia Mozzicato. 2009. MedDRA: An overview of the medical dictionary for regulatory activities. *Pharmaceutical Medicine*, 23(2):65–75.

Karen O'Connor, Abeed Sarker, Jeanmarie Perrone, and Graciela Gonzalez-Hernandez. 2020. Promoting reproducible research for characterizing nonmedical use of medications through data annotation: Description of a twitter corpus and guidelines. *Journal of Medical Internet Research*, 22(2):e15861.

Jeffrey Pennington, Richard Socher, and Christoper Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Luca Soldaini and Nazli Goharian. 2016. QuickUMLS: a fast, unsupervised approach for medical concept extraction. In *SIGIR Workshop on Medical Information Retrieval (MedIR)*, pages 1–4.

V.G.Vinod Vydiswaran, Grace Ganzel, Bryan Romas, Deahan Yu, Amy Austin, Neha Bhomia, Socheatha Chan, Stephanie Hall, Van Le, Aaron Miller, Olawunmi Oduyebo, Aulia Song, Radhika Sondhi, Danny Teng, Hao Tseng, Kim Vuong, and Stephanie Zimmerman. 2019. Towards text processing pipelines to identify adverse drug events-related tweets: University of Michigan @ SMM4H 2019 Task 1. In *Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*, pages 107–109, Florence, Italy, August. Association for Computational Linguistics.

Davy Weissenbacher, Abeed Sarker, Arjun Magge, Ashlynn Daughton, Karen O'Connor, Michael J. Paul, and Graciela Gonzalez-Hernandez. 2019. Overview of the fourth social media mining for health (SMM4H) shared tasks at ACL 2019. In *Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*, pages 21–30, Florence, Italy, August. Association for Computational Linguistics.

Xinyan Zhao, Deahan Yu, and V.G.Vinod Vydiswaran. 2019. Identifying adverse drug events mentions in tweets using attentive, collocated, and aggregated medical representation. In *Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*, pages 62–70, Florence, Italy, August. Association for Computational Linguistics.