


Identifying Tweets with Personal Medication Intake Mentions using Attentive Character and Localized Context Representations


Jarashanth Selvarajah

(Postgraduate Institute of Science, University of Peradeniya, Peradeniya, Sri Lanka

 <https://orcid.org/0000-0002-3228-0145>, jarashanth02@gmail.com)

Ruwan Nawarathna

(Department of Statistics and Computer Science, University of Peradeniya, Peradeniya, Sri Lanka

 <https://orcid.org/0000-0001-5843-8919>, ruwand@sci.pdn.ac.lk)

Abstract: Individuals with health anomalies often share their experiences on social media sites, such as Twitter, which yields an abundance of data on a global scale. Nowadays, social media data constitutes a leading source to build drug monitoring and surveillance systems. However, a proper assessment of such data requires discarding mentions which do not express drug-related personal health experiences. We automate this process by introducing a novel deep learning model. The model includes character-level and word-level embeddings, embedding-level attention, convolutional neural networks (CNN), bidirectional gated recurrent units (BiGRU), and context-aware attentions. An embedding for a word is produced by integrating both word-level and character-level embeddings using an embedding-level attention mechanism, which selects the salient features from both embeddings without expanding dimensionality. The resultant embedding is further analyzed by three CNN layers independently, where each extracts unique n-grams. BiGRUs followed by attention layers further process the outputs from each CNN layer. Besides, the resultant embedding is also encoded by a BiGRU with attention. Our model is developed to cope with the intricate attributes inherent to tweets such as vernacular texts, descriptive medical phrases, frequently misspelt words, abbreviations, short messages, and others. All these four outputs are summed and sent to a softmax classifier. We built a dataset by incorporating tweets from two benchmark datasets designed for the same objective to evaluate the performance. Our model performs substantially better than existing models, including several customized Bidirectional Encoder Representations from Transformers (BERT) models with an F1-score of 0.772.

Keywords: Drug surveillance; character-level embedding; context-aware attention; convolutional neural networks, bidirectional gated recurrent units

Categories: I.2.7, I.5.1, M

DOI: 10.3897/jucs.84130

1 Introduction

The unprecedented growth of social media platforms brings an abundance of user-generated content on various topics. Social networks form a platform for people to share their opinions, insights, experiences, and perspectives. More users have engaged in both generic social networks (such as Twitter, Facebook, to name a few) and in medical forums (such as DailyStrength, MedHelp, and others) to gather health-related information, to share their experiences on diseases, symptoms and treatments or to interact with others

facing similar problems. User-generated drug-related chatter on social media constitutes a valuable resource for post-marketing drug surveillance, also known as pharmacovigilance [Sarker and Gonzalez 2015]. With the advancement of processing a large volume of these data automatically, using natural language processing (NLP) and deep learning, new opportunities have opened for conducting public health monitoring and surveillance. Several studies have mined drug-related social media posts to detect various medical abnormalities such as Adverse Drug Reactions (ADRs) [Ding et al. 2018] and medication abuse [Sarker et al. 2016]. However, the content used in these studies is the aggregation of social media data that mentions a drug without considering whether the user has consumed it. Without this knowledge, a valid assessment of the effects of medication intake cannot be undertaken. To solve that, we need a system that can automatically distinguish posts that express personal intake of medicine from those that do not. To carry out our experiments, we use Twitter as the social media channel due to its vast reach and large user base. The objective of this study is to distinguish drug-related tweets into personal health mentions and others by leveraging various deep learning technologies.

The use of social media for health monitoring and surveillance introduces various NLP challenges, including misspellings, informal sentences, descriptive medical concepts, idiomatic expressions and ambiguity. Twitter poses additional challenges such as brevity, noisiness (i.e., URLs and promotional advertisements), unusual structure and data imbalance. It has also been studied that traditional NLP methods that are applied to longer texts are inadequate when applied to shorter texts such as tweets [Sarker and Gonzalez 2015]. Moreover, the distinction between personal intake mentions and other mentions is vague as it is often difficult to understand from the informally expressed tweets whether a user has actually consumed a medicine or just mentioned it.

The task of identifying posts mentioning personal intake of medicine from Twitter was introduced at Social Media Mining for Health (SMM4H) shared task 2017 [Sarker and Gonzalez 2017] and resumed in SMM4H 2018 [Weissenbacher et al. 2018]. The objective was to categorize a drug-related tweet into one of three classes: personal medication intake, possible medication intake, and non-intake. Tweets that clearly express personal medication intake are included in personal medication intake class. The possible medication intake class consists of tweets that are ambiguously expressed but suggest that the user might have taken the medication and non-intake class contains tweets that mention medication names but do not indicate personal intake. Participants have tried both traditional statistical methods and deep learning models. The best traditional classification method proposed by NRC-Canada implements a support vector machine classifier using a variety of surface-form, sentiment, and domain-specific features [Kiritchenko et al. 2018]. InfyNLP explores the applicability of various deep learning models and concludes that a stacked ensemble of shallow CNN performs relatively better when hyperparameters of the model are fine-tuned [Mahata et al. 2018]. This model was ranked first in the SMM4H 2017 shared task. The system that got first place in the SMM4H 2018 applies word-level stacked BiLSTM with context-aware attention [Xherija 2018].

In SMM4H 2019, a binary classification task for the identification of personal health mentions was conducted. This task intended to build models to identify personal health mentions in a generalized context rather than medication only context [Weissenbacher et al. 2019]. Team UZH developed a BERT-based ensemble system that achieved the best F1-score among other participated systems [Ellendorff et al. 2019]. Team ASU-NLP built a model using BioBERT and feedforward neural network, which obtained the second-highest F1-score [Gondane 2019].

Besides SMM4H shared tasks, Jiang et al. constructed a personal experience tweets (PET) dataset related to the effects caused by four dietary supplements using a bootstrap

method with a machine learning-based filter [Jiang et al. 2016]. Three classifiers, namely, decision tree, KNN, and neural network, were used to filter out off-topic tweets via majority voting. Although a small portion of tweets is PETs, the bootstrap approach efficiently improves the balance of two class dataset with a reduced amount of annotations. In another study, Jiang et al. employed the same ensemble of machine learning-based classifiers to identify the PETs from the same dataset [Jiang et al. 2017]. Moreover, the PETs were further annotated to extract the potential dietary supplement effects (adverse and beneficial) which were mapped to clinical concepts with SNOMED CT (Systematized Nomenclature of Medicine – Clinical Terms) using an open-source tool called medpie [Benton et al. 2012]. Jiang et al. constructed another PET drug-related dataset and applied word embedding based LSTM model to identify the personal mentions [Jiang et al. 2018]. The same study later experimented with four word embedding techniques: GloVe, word2vec, fastText, and wordRank [Narui et al. 2020].

Karisani et al. developed a dataset consisting of tweets across six diseases and conditions [Karisani and Agichtein 2018]. Tweets were manually annotated as four categories: self-mention, other-mention, awareness and non-health. They built a model named WESPAD by combining lexical, syntactic, word embedding-based, and context features and feeding them into a logistic regression classifier. In their experiments, they combined self-mention and other-mention tweets as positive, and awareness and non-health tweets as negative. The authors declared that their model requires relatively little training data which might be a good sign for adapting the model for new diseases and conditions.

Generally, deep learning models suffer from the lack of manually annotated datasets and class imbalance. The work presented in [Zhu and Jiang 2021] has investigated the impact on classification performance in predicting tweets mentioning personal medication intake mentions using two semi-supervised learning models, with different mixes of labelled and unlabelled data in the training set.

Only a limited number of studies have investigated the problem of identifying personal health mentions from Twitter data. The proposed models have limited potential in addressing intrinsic characteristics of tweets, such as excessive Out-of-Vocabulary (OOV) words and irregular structure. This drawback is addressed by adding character-level features to the conventional word embedding and combining them with an attention mechanism. Unlike other studies, we use different levels of n-grams to encode the context better, which is challenging in tweets as they are short.

Deep learning has produced breakthroughs in many fields, such as computer vision, speech recognition, machine translation, and text classification, in recent years. There are two widely used deep learning models for text classification, namely, convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs are generally used in computer vision; however, they have given promising results on various NLP tasks. CNN can learn the spatial relationship between adjacent words but is inadequate to learn sequential correlations [Liu and Guo 2019]. On the contrary, RNNs are specialized for processing sequential data but unable to extract spatial features. RNNs are widely used in text classification due to their ability to capture contextual information by maintaining a state of all the previous inputs. However, for long data sequences, vanilla RNNs cause problems, specifically, exploding and vanishing gradients. Long short-term memory (LSTM) and Gated Recurrent Unit (GRU) are variants of RNN that effectively solve vanishing gradient and gradient explosion problems. Moreover, they are much better at capturing long-term dependencies than vanilla RNNs. Unlike LSTM, Bidirectional long short-term memory (BiLSTM) network exploits long-range context information in both forward and backward directions. Thus, BiLSTM is found to be well suited for

sequential data processing than LSTM. Moreover, various attention mechanisms have been used widely in order to enforce the inputs that contribute more to the end goal (e.g., classification) due to the fact that not all the inputs contribute equally.

Deep learning models generally use pre-trained word embedding for the vector representation of the text. Word embedding is a distributed representation of words as low-dimensional vectors where semantically similar words have similar representations. Due to the characteristics of social media text (e.g., informal spellings and abbreviations), a large number of words do not have corresponding word vectors. Hence, those OOV words are randomly initialized to some specific values, which often cause misclassification.

Our primary contribution in this study is to propose a novel deep learning architecture to facilitate the identification of personal medication intake mentions from social media text. The proposed deep learning architecture incorporates character-level and word-level embedding using an attention mechanism together with convolutional neural networks (CNNs), bidirectional gated recurrent units (BiGRU), and a penultimate-level attention mechanism. To alleviate misclassifications caused by OOV words, we encode each token using both word-level and character-level embedding. Besides, we employ an embedding-level attention mechanism to select the important features from both embeddings.

The rest of this paper is organized as follows. In section 2, we explain our proposed model in detail. Section 3 presents the experiments conducted to evaluate the performance of the model and summarizes the results obtained. Section 3 also incorporates a discussion that includes performance comparison with similar models. Finally, a few concluding remarks that highlight the major findings of the study are given in section 4.

2 Methodology

The proposed method is developed based on state-of-the-art deep learning-based natural language processing techniques. The details of the dataset and steps of the methodology are provided in the subsections below.

2.1 Datasets

A limited number of benchmark datasets are available for identifying tweets with personal medication intake mentions; however, only tweet ids referring to them are provided due to Twitter's privacy policy. Reproducing an exact copy of a dataset is impractical because (a) many tweets are not downloadable since either the tweets have been removed or the accounts associated with the tweets are no longer active; (b) only partial datasets are available for the public. We created a dataset by combining tweets from the following two datasets. The first dataset is obtained from the official website for the SMM4H shared task 2018 [Weissenbacher et al. 2019]. In this dataset, drug-related tweets were labelled into one of three classes: personal intake—clearly expressed personal medication intake mentions, possible intake—ambiguously expressed personal medication intake mentions and non-intake—no indications of personal intake mentions. A comparison between the original and downloaded tweets of the dataset released by SMM4H is displayed in Table 1. The second dataset is also a drug-related twitter dataset, released publicly by [Jiang et al. 2018], where tweets have been annotated for the presence of personal medication intake mentions as a binary classification problem. The statistics of the original and downloaded tweets of this dataset are shown in Table 2.

We discarded tweets with possible intake mentions from the first dataset and combined the rest with the second dataset to form a new dataset. We name this resultant

Tweets	Class 1 (personal intake)	Class 2 (possible intake)	Class 3 (non-intake)	Total
Original	3,683	5,916	8,174	17,773
Downloaded	3,158	5,002	6,953	15,113

Table 1: The statistics of the original and downloaded tweets of the dataset released by SMM4H. Classes 1, 2 and 3 represent personal intake, possible intake, and non-intake tweets, respectively.

Dataset Type	Positive	Negative	Total
Original	2,962	9,369	12,331
Downloaded	1,742	5,725	7,467

Table 2: Statistics of the original and downloaded tweets of the second dataset released publicly by [Jiang et al. 2018]. Positive and negative classes represent personal medication intake mentions and other mentions.

dataset as Personal Medication Intake (PMI) dataset. Moreover, we filter out tweets that have less than five tokens. We divide the PMI dataset as train set, validation set and test set, where each holds a portion of 60%, 20%, 20%, respectively. We keep the positive to negative ratio consistent between all three splits to get a more realistic performance. Table 3 summarizes the statistics of the PMI dataset.

Class	Train	Validation	Test	Total
Positive	2,900	967	967	4,834
Negative	7,091	2,364	2,364	11,819
Total	9,991	3,331	3,331	16,653

Table 3: Statistics of the new PMI dataset with train, validation and test splits.

Table 4 lists down an excerpt of tweets from the PMI dataset. Class 1 and 0 represent personal medication intake and non-intake mentions, respectively. Tweet Nos 1 and 2 clearly indicate they are personal medication intake mentions. Although tweet No. 3 states a ground truth, due to the nature of Twitter, it is more likely to be a personal medication intake mention. As tweet No. 4 states, sharing someone else’s experience is not considered a personal health mention. Tweet No. 5 is about medication but neither shares a personal experience nor a third-party’s experience, thus labelled as non-intake. A person suggested a medication that can be taken during winter in tweet No 6, but it is unclear whether the person might have taken it earlier, hence labelled as non-intake. In our proposed model, we propose techniques to handle such a diverse set of tweets.

2.2 Data Preprocessing

The raw tweets, collected by querying the Twitter API, have noise in terms of user mentions (e.g., @xyz), URLs (e.g., http://www.xzy.com), elongations (e.g., yessss, soooo), incorrect/informal spellings (e.g., gr8, lovin), abbreviations (e.g., idk for “I don’t

No	Tweet	Class
1	“then i took 2 ibuprofens & i still have a headache, nothing is working for me”	1
2	“that tylenol really worked”	1
3	“daydreams are what happen when you take tylenol pm in the am”	1
4	“tylenol is a mom’s best friend”	0
5	“please tell me why every patient you admit needs to have albuterol 1.25 mg tid?!”	0
6	“people are going insane about this snow coming. take a xanax please!!!!”	0

Table 4: Sample tweets from the PMI dataset.

know”), emoticons, and so on. We adopted the following preprocessing steps to cope with the noisy nature of tweets.

We normalized URLs to the word ‘URL’ and user ids to the word ‘USER_MENTION’. Also, we retained the text of hashtags by removing only the pound sign (#) as hashtags hold some useful information (i.e., a trending topic). Users often use emoticons to convey different emotions. We selected a set of frequently used emoticons and seek for their presence in tweets. If a match is found, we replaced it as either ‘EMO_POS’ (i.e., positive emotion) or ‘EMO_NEG’ (i.e., negative emotion) based on the polarity of the emoticon. To address elongations, we shortened the character repeated more than 2 times to 2 times (e.g., funnnny to funny). Besides, punctuations and excessive white spaces were removed, and the tweets were lowercased.

2.3 Model Architecture

Figure 1 shows the overall structure of our model. We implement a character embedding layer and a word embedding layer to obtain character-level and word-level representations for each token, respectively. An embedding-level attention mechanism extracts the vital information from both layers and generates a resultant embedding independently analyzed by four sub-modules. Sub-module 1 includes a BiGRU layer which incorporates a forward GRU layer and a backward GRU layer to learn the context information for each token in the input sequence based on both the preceding and succeeding tokens. An attention layer further analyzes the concatenated token-level hidden units of the BiGRU layer to give more weights to the deciding tokens. Sub-modules 2, 3 and 4 are similar to Sub-module 1; however, each includes a one-dimensional CNN layer before BiGRU layer, where unigrams, bigrams, and trigrams are extracted, respectively. The attention scores of each sub-module are summed up and forwarded to the classification layer. The justifications for adding each element and its inner workings are discussed below in detail.

2.3.1 Character Embedding Layer

This layer exhibits an embedding vector for each word by feeding its character sequence into a BiGRU model. First, the dataset is segmented to identify the length of the sentence with the maximum number of tokens (maxlen_s) and the length of the token with the maximum number of characters (maxlen_t) and to list down all the unique characters

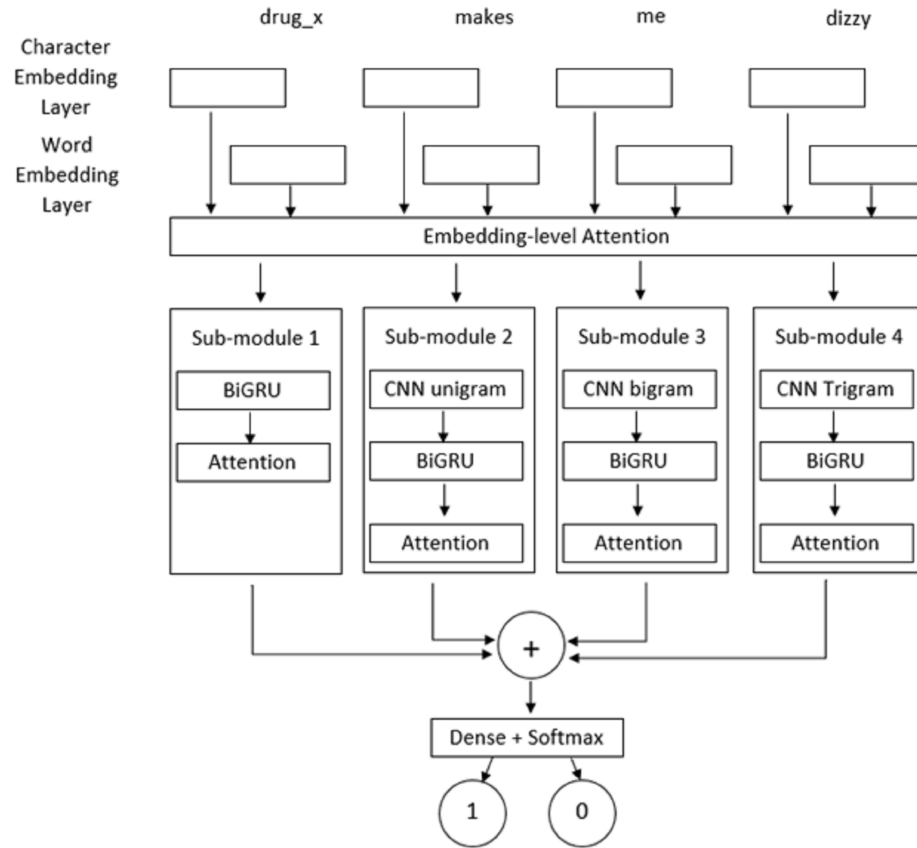


Figure 1: Architecture of our proposed model. The model consists of a character embedding layer, word embedding layer, embedding-level attention, four BiGRU-based Sub-modules

that are present in the entire corpus and their frequencies. Then, the characters are sorted from the most common to the least and are mapped to integers starting from 1. Next, each character in the dataset is mapped to its corresponding integer. After that, sentences and tokens are zero-padded to the length of `maxlen_s` and `maxlen_t`, respectively. Hence, the input dimension of this layer is $(batch_size, maxlen_s, maxlen_t)$, where `batch_size` refers to the number of samples utilized in one iteration. The input is fed into an Embedding layer where each character is initialized with a 100-dimensional vector. The ‘trainable’ parameter of this embedding layer is set to ‘True’ so that the model can learn the embedding weights while training. A character-level embedding for each word is obtained from its character sequence by concatenating the last hidden states of the forward and backward GRUs. The number of features in the hidden state of GRUs is set to 100, so we get a 200-dimensional embedding. Since an embedding level of attention is applied to both character-level and word-level embedding, it is required that the dimensionalities of both should be consistent. Figure 2 explains how a character-level embedding is formed for the word ‘sore’.

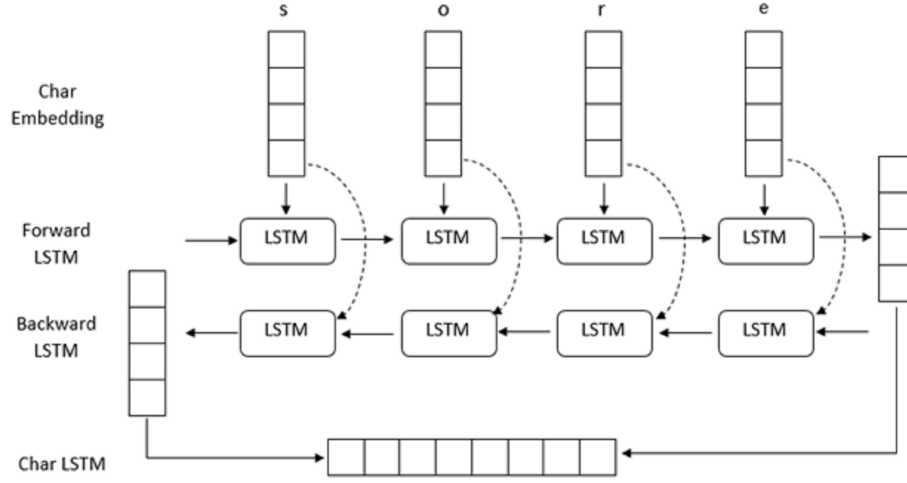


Figure 2: Character-level embedding generation for the word ‘sore’ by feeding its characters to a BiGRU model. Each character in the character vocabulary is randomly initialized to a 100-dimensional vector.

2.3.2 Word Embedding Layer

Word embedding layer maps each token to a real-valued vector of some fixed dimension. Let d be the dimension of word embedding, and l be the length of a tweet. The vector representation of the tweet is encoded by an embedding matrix $C \in R^{d \times l}$, where $C_i \in R^d$ corresponds to the word embedding of the i^{th} token in the tweet. It has been observed that treating the embedding as fixed constants performs better than setting them as learnable model parameters. We use a publicly available pre-trained word embedding, known as GloVe [Penninton et al. 2014] to obtain word vectors for the tokens. Distributed word embedding, which is seemingly used in many NLP applications to represent words, is well known for capturing the semantic relationship between words. To initialize our model, we adopt a 200-dimensional GloVe embedding built over 2 billion tweets with a vocabulary of 1.2 million unique words.

2.3.3 Embedding-Level Attention Layer

This layer intends to apply an embedding level of attention [Vaswani et al. 2017] to combine character-level embedding with word embedding. Using word embedding alone to represent words does not address OOV words and fails to take the fine-grained character features into account. Moreover, simple concatenation of character-level and word-level embeddings adds character level features but expands the dimensions of word embeddings without giving them any weights. Weighted integration of character-level and word-level embeddings can help the model to dynamically determine how much information to use from the character-level and word-level embeddings. The neural architecture of the embedding level attention mechanism is expressed in Eq. 1 and Eq. 2.

$$a = \sigma(V_a * \tanh(W_a * x + U_a * q)) \quad (1)$$

$$\bar{x} = a.x + (1 - a).q \quad (2)$$

In Eq. 1, V_a , W_a and U_a are weights to calculate the attention matrix a , and σ denotes the sigmoid operation. In Eq. 2, x and q represent word-level and character-level embeddings, respectively. The attention matrix a decides the percentage of information to be used from each embedding. \bar{x} is the resultant embedding after the embedding level attention is applied.

2.3.4 One-dimensional convolutional layer

Our model employs a one-dimensional convolutional operation on a single convolutional layer to capture local relationships between adjacent words (i.e., extract local n-gram features) in a tweet. Let l and d be the length of the tweet and word vector, respectively. Let $X \in R^{d \times l}$ be the embedding matrix of the tweet. A convolution operation involves a filter with weights $W \in R^{d \times w}$, which is applied to a window of w words to get a feature mapping $c \in R^{l-w+1}$. For instance, i -th element of the feature map is generated from a window of words $X[i : i + w - 1]$ by Eq. 3.

$$c_i = f \left(\sum (X[i : i + w - 1] \circ W) + b \right) \quad (3)$$

In Eq. 3, $b \in R$ is a bias vector, and $f(x)$ is a non-linear activation function, generally rectified linear units (ReLU). \circ is the element-wise product between two matrices. We convolve the filter over all word windows of the tweet and extract the n-grams feature vector (i.e., a feature map) of size $l-w+1$ as shown in Eq. 4. It is to be noted that we do not use any pooling operation.

$$c = [c_1, c_2, \dots, c_{l-w+1}] \quad (4)$$

We apply a set of 200 independent filters of sizes $1 \times d$, $2 \times d$, $3 \times d$, on Sub-modules 2, 3, and 4, respectively, to get a 200-dimensional vector for each ngram (Eq. 5). Hence, all three convolutional layers have 200 filters; however, each has a different kernel size.

$$C = [c^1, c^2, \dots, c^{200}] \quad (5)$$

2.3.5 BiLSTM and BiGRU Layers

In our method, two powerful variants of recurrent neural networks (RNN), specifically, long short-term memory (LSTM) and gated recurrent unit (GRU), are proposed to overcome the shortcomings of standard RNN [Goodfellow et al. 2016].

The standard LSTM network can only exploit the historical context, which may be inadequate to capture meaningful details. Bidirectional long short-term memory (BiLSTM) utilizes both the preceding and succeeding contexts by concatenating forward hidden state \vec{h}_t and backward hidden state \overleftarrow{h}_t (i.e., $h_t = \vec{h}_t + \overleftarrow{h}_t$). Gated Recurrent Unit is a simplified version of LSTM; thus, GRU is slightly faster than LSTM. The motivation behind the bidirectional gated recurrent unit (BiGRU) remains the same as BiLSTM, except replacing the LSTM nodes with GRU.

In the proposed model, the preceding convolutional layer extracts local contextual information for each word in a tweet but do not capture the long-term sequential relationship and the dependency among them. Therefore, we employ a BiGRU layer to extract

the sequential information from the feature sequences obtained by the convolutional layer. The forward *GRU* (denoted as \overrightarrow{GRU}) and the backward *GRU* (denoted as \overleftarrow{GRU}) of the BiGRU layer read the feature sequences from c_1 to c_{l-w+1} and c_{l-w+1} to c_1 , respectively. The outputs of the BiGRU are defined as given in Eq. 6 and Eq. 7.

According to Eq. 6 and Eq. 7, the annotation for a given feature sequence c_n is obtained by concatenating the forward hidden state \overrightarrow{h}_f and the backward hidden state \overleftarrow{h}_b .

The hidden state dimension of the character-level BiGRU layer is set to half the length of the word-level embedding to ensure that we get an embedding consistent with word-level embedding by combining the last hidden states of both forward and backward GRUs. BiGRU holds a 256-dimensional memory unit.

$$\overrightarrow{h}_f = \overrightarrow{GRU}(c_n), n \in [1, l - w + 1] \quad (6)$$

$$\overleftarrow{h}_b = \overleftarrow{GRU}(c_n), n \in [l - w + 1, 1] \quad (7)$$

2.3.6 Attention Layer

Although BiGRU can effectively learn the sequential correlations between tokens in both forward and backward directions, the mechanism does not distinguish the important words that contribute more to the meaning of a sentence. First-person singular pronouns (e.g., I, me, my, mine) are more likely to emphasize personal mentions, whereas stopwords (e.g., the, at, on) are not. The attention mechanism assigns weights to each token based on its importance. The weight is a probability distribution among all the tokens within a tweet. The weights are multiplied using matrix multiplication with the summed hidden states of both forward and backward GRUs.

As discussed in subsection 2.3.5, the BiGRU network learns a vector representation for each word by exploiting its preceding and succeeding tokens but fails to account that not all the words equally contribute to determine the meaning of a word. This drawback is resolved by adding an attention layer to BiGRU where a weighted sum of the context words are used to form a vector for each word. The ‘attention’ assigns higher weights to the context words, which contribute significantly to the meaning of a word while keeping the weights lower than the rest.

2.3.7 Output Layer

The features generated from the attention layer are passed to a fully connected layer of size two with softmax activation, whose output is the probability distribution over the two classes (See Figure 1).

2.4 Model Optimization and Hyperparameters

To calculate the loss of the model while training, we have a choice of using either Binary Cross Entropy Loss [Goodfellow et al. 2016] or Cross Entropy Loss [Goodfellow et al. 2016]. Our preliminary experiments revealed that the performance of Cross Entropy Loss was slightly better than Binary Cross Entropy Loss. Hence, we use Cross Entropy Loss for the rest of the experiments. Although both apply negative log-likelihood loss, the model’s classification layer needs to be modified to accommodate a specific loss function. Binary Cross Entropy Loss needs a single neuron with a sigmoid activation, whereas Cross

Entropy Loss requires two neurons with a softmax activation. Also, gradient clipping is applied to keep the gradients within a certain threshold, thus avoiding gradient explosion. The weights are updated after each batch is processed, using Adam optimization algorithm [Goodfellow et al. 2016]. We impose “early-stopping” regularization that stops training when the validation loss no longer reduces to avoid over-fitting. We also apply dropouts with a rate of 0.3 to the word embedding layer and the BiGRU layers. Dropout is a method of regularization that randomly drops out certain connections to the next layer to force the model to learn along different paths. The model was built by gradually adding salient components.

The challenge was to find the right features and the proper networking of the model. Hyperparameters were manually adjusted to reach optima. The configurations of all hyperparameters used in CNN and BiGRU layers are summarized in Table 5. The batch size, number of epochs and learning rates are set to 10, 39 and 0.0001, respectively.

Hyperparameters	Value
Word embedding size	200
Character embedding size	100
Character level GRU’s hidden state size	0.5 * size of the word embedding’s dimension
Number of CNN filters	200 for each CNN layer (total 3 layers)
Kernel size	1/2/3
Padding	valid
BiGRU’s hidden state size	256
Dropout	0.3
Batch size	10
Epoch	30
Learning rate	0.0001

Table 5: Hyperparameters settings of the CNN and BiGRU models.

3 Results and Discussion

This section explains the details of our experiments to evaluate our model and presents the results obtained. The model is implemented in PyTorch [Paszke et al. 2019] and Keras [Chollet et al. 2015], and trained on Google Colab Pro with GPU utilization. Since we used Google Colab Pro, every time a new virtual machine instance is created, computations were not deterministic, as reproducible results are not guaranteed across different platforms. In order to make computations deterministic to a certain level, we manually seeded all the libraries which use random numbers with a fixed value. Moreover, we examined deterministic algorithm implementations in cuDNN [Paszke et al. 2019]. However, due to the prolonged training time of deterministic algorithms, we ran each model 5 times using non-deterministic algorithm implementations in cuDNN and reported the average results.

3.1 Evaluation Metrics

Since our study is defined as a binary classification problem, we report precision, recall, and F1-score for the positive class defined as the evaluation metrics based on the confusion matrix. The confusion matrix given in Table 6 shows the definitions of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). Recall is the percentage of correctly classified positive instances and it is defined as $TP/(TP+FN)$. Precision computed from $TP/(TP+FP)$ is the percentage of correctly classified positive instances from the predicted positives. F1 score is defined as the harmonic means of the precision and the recall of the model.

Actual Class	Predicted Class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Table 6: Confusion matrix which defines True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) of a binary classification model.

3.2 Ablation Analysis on Our Proposed Model

Table 7 presents the results of an ablation analysis conducted to elucidate the relative contribution of each component to our proposed model. As mentioned in Section 2.1, our train-validation-test split of the PMI dataset was 60-20-20. Our proposed model (M1) (See Figure 1 for the model architecture) gives an F1-score of 0.772 on the PMI dataset described in subsection 2.1. To address class-imbalance, we weigh the positive class with a quotient calculated by dividing the total number of negative instances by the total number of positive instances in the training set. A vanilla BiGRU model (M9) is declared as our baseline in order to evaluate how well our model performs. As shown in Table 7, our model shows 4.6% improvement in terms of F1-score compared to the baseline. The results further reveal that each component plays a vital role in performance gain, which is discussed in detail in the following subsections. The contribution of a constituent to the overall performance is evaluated by dropping it and report the performance of the residual model.

3.2.1 The Impact of Character-level Embedding

There is a 2% drop in F1-score when character-level embedding is removed from the proposed model (M2). This can be attributed to the following limitations of the word embedding. The context-independent behavior inherent to the word embedding may not discern polysemous words. Moreover, the word embedding eschews the use of character-level features, and thus fails to properly represent an incorrectly spelled or an out-of-vocabulary (OOV) word.

To alleviate the constraints caused by the word embedding to our model, we include a character-level embedding along with word-level embedding, which can normalize a considerably large number of OOV tokens present in tweets efficiently, based on their character sequences.

Model	Model	Precision	Recall	F1
M1	Proposed Model	0.764	0.781	0.772
M2	Proposed Model – Char Embedding	0.796	0.704	0.747
M3	Proposed Model with embedding level concatenation	0.744	0.770	0.757
M4	Proposed Model with embedding level summation	0.690	0.824	0.751
M5	Proposed Model – All CNN layers + CNN with filter size = 3	0.746	0.764	0.755
M6	Proposed Model – All CNN layers	0.708	0.792	0.748
M7	Proposed Model – Sub-module 1	0.670	0.850	0.749
M8	Proposed Model – penultimate level attention (Sub-modules 2, 3 and 4 are discarded)	0.716	0.755	0.735
M9	BiGRU (baseline)	0.691	0.765	0.726

Table 7: Ablation study results of the proposed model and its residuals. The minus sign indicates that the mentioned Component/Sub-module is discarded. For example, M2 represents the proposed model without character-level embedding.

Even though the character-level embedding is incorporated, the performance gain highly depends on how information from both character-level and word-level embeddings is compiled. To support the argument, we report the results of a variant of our proposed model where both embeddings are concatenated together as model M3. The trivial difference between models M2 and M3 interprets that adding a character-level embedding by concatenation comes at the cost of dimensionality increase, thereby perplexing the model to learn and converge. On the other hand, we experimented by summing the vectors from both embeddings (model M4). Although this approach does not expand dimensionality, summing embeddings without giving them any weight causes inconsiderable performance gain. With the embedding-level attention, it is more evident that our model successfully learns the salient features from both embeddings dynamically without expanding its dimensionality. This can be inferred from the performance disparity between the models M1 and M2.

3.2.2 The Impact of Localized Context Representations

Even though CNN is a powerful tool to extract local context features (i.e., ngrams), improper placement of it in the network would cause performance reduction. As illustrated in Figure 1, we extract the localized context representations with the help of four Sub-modules. We did several experiments by placing CNNs in various positions with different settings; however, most showed relatively low impact on the overall performance. The proposed model includes three CNN layers where each extracts an ngram feature (unigrams, bigrams, and trigrams) from the resultant embedding. These features are further encoded by BiGRUs individually. Meanwhile, the resultant embedding is also encoded by a BiGRU directly without going through a CNN layer (Sub-module 1).

The intuition behind this setup is discussed as follows. The first experiment included only a single CNN layer (i.e. sub-modules 2 and 3 are omitted) with filter size of three

to extract trigrams and forwarded its output to the BiGRU layer. This model is defined as M5. The results revealed that the performance of the model was substantially lower compared to our proposed model (model M1). This discrepancy can be rationalized due to two reasons: (a) tweets are usually shorter in length, and thus converting to trigrams would lose more information. (b) noisy and colloquial nature of tweets with frequent misspellings and excessive use of abbreviations, thereby compacting tokens would not work well. As a result, we extract all unigrams, bigrams, and trigrams using CNN layers and process them independently.

The performance difference between model M6 and our proposed model M1 sheds light on the contribution of including the CNNs. Convolution operations capture local context features to a certain level; however, filter-based summarization often leads to lose some original information, and hence we encode them using BiGRUs in Sub-module 1. Model 7 performs slightly lower than the proposed model, which asserts the above-mentioned statement.

3.2.3 The Impact of the Attention Mechanism

Even though LSTM effectively learns representations for variable-length sequences, it fails to account for future context. The BiLSTM network addresses this limitation by considering both historical and future context. Our preliminary studies showed that both BiLSTM and BiGRU performed to a similar degree. Since BiGRU has fewer parameters than BiLSTM, we chose BiGRU over BiLSTM to speed up calculations.

Another drawback of LSTM is that the later words are more dominant than the earlier words. This is because the internal gating mechanism forgets information as the sequence grows, leading to a vanishing gradient. This problem is addressed by the utilization of both forward and backward contexts, yet it is not a viable solution. In addition, not all the words equally contribute to the classification. To exemplify this, let's look at the two sentences: (a) "I need Tylenol". (b) "I need more Tylenol". The first sentence does not mention personal intake, whereas the second sentence does. The word "more" changes the classification output from negative to positive. The attention layer gives more weight to the decisive words by assigning a probability distribution among words. The performance difference between M8 and exhibits the potential of the attention mechanism.

3.2.4 Error Analysis and Misclassification

We observe that the use of colloquial language and ambiguous statements are two of the key reasons for the performance loss. Personal health mentions are often written in descriptive text rather than using standard medical terminologies. An experience such as headache can be expressed in multiple ways, thereby perplexes a model to generalize. Sometimes, it is hard to validate a mention as personal even by humans. For example, in a sentence "I got medication X so that I can sleep well tonight", it could be either the person has taken the medication in the past or just the first time taking it. If it is the first time the person takes the medication, it would not be considered a personal health experience. Another major concern was the presence of more drug-related generic statements over personal experiences. Users often mention general information and do not state personal experiences. For example, the sentence "drug x causes trauma" is a generic statement than a personal experience.

Further, the twitter posts are short and thus contain very little information. Posts often consist of a large proportion of spelling errors. If these OOV words are not meaningfully

parsed, it will lead to misclassification. As in any other dataset, noisy data was also an issue. The noisy data includes advertisements, posts with usernames identical to drug names, and expressions that do not represent a medical condition (e.g., ‘heart attack’ is often used to emphasize shocking events).

A large portion of the PMI dataset exhibits multiple impediments, but our proposed model copes with some of these constraints very well.

3.3 Performance Comparison with Customized BERT Models

In this subsection, we analyze the behavior of the BERT and its variants [Ellendorff et al. 2019] to learn how well they perform on the PMI dataset. It is recalled that the BERT model is trained on Wikipedia and book corpus, and thus not exposed to vernaculars.

The transformer network is designed as an encoder-decoder architecture where the encoding component is a stack of six encoders. In contrast, the decoding component is a stack of decoders of the same number. All the encoders are identical in structure. The decoders are also identical in structure but slightly differ from the encoders. Each encoder has a self-attention layer and a feedforward network. Each input token is fed to the first encoder, and the output of the first encoder is forwarded to the next encoder, and so on. Each decoder has a self-attention layer, encoder-decoder attention layer and a feedforward network. The output of the last encoder is sent to the first decoder. The transformer network has been trained on a large corpus as a unidirectional language model, and the model is publicly available. The trained model can be fine-tuned for various downstream tasks (i.e., transfer learning in NLP). More importantly, since the transformer network does not need to process the data sequentially, the model accommodates parallelization.

BERT uses the encoder part of the transformer and trains it as a bidirectional language model on a large Wikipedia and book corpus utilizing a technique known as “masked language model”, where a word is masked, and the masked word is predicted using the past and future contexts. BERT can also be fine-tuned for downstream tasks. BERT provides a set of pre-trained models trained under distinct settings.

We construct four customized BERT models using the embedding provided by the ‘bert-base-uncased’ model implementation in the ‘pytorch-pretrained-bert’ library [Paszke et al. 2019]. We compare the performances of these models against our proposed model. The architecture of the models is described below.

An additional BERT specific token, referred to as [CLS], is inserted at the start of each tweet along with the preprocessing steps explained in subsection 2.2. BERT segments tokens that are not available in its vocabulary into sub-tokens using a WordPiece tokenizer. BERT delivers two types of embeddings: an individual vector representation for each token and a single summarized vector for each input sequence, also known as ‘pooled output’, which corresponds to the vector representation of its [CLS] token. The first model uses the ‘pytorch-pretrained-bert’ library’s ‘BertForSequenceClassification’ module by setting ‘num_labels’ parameter to 2 and the logits are sent to a softmax layer for classification. The second model utilizes the ‘pooled output’ features of the ‘pytorch-pretrained-bert’ library’s ‘BertModel’ module. The outputs are further analyzed by a 3-layer neural network. The third model incorporates a BiLSTM layer, processes the ‘sequential outputs’ of the ‘BERTmodel’ module, and forwards the output to a classification layer. The last model is also similar to the third model, except that the BiLSTM layer is followed by an attention layer.

Table 8 displays the performance metrics of the customized BERT models. In the default BERT model, the final hidden state of the first word “[CLS]” from BERT is input to a softmax classifier, and the entire model is fine-tuned. The results were slightly higher

than a vanilla BiGRU model. Although BERT gives a context-sensitive representation for each token, encoding an OOV token, especially an incorrectly spelt one, with predefined sub-tokens will not always work for a dataset with strange characteristics. Model B2 includes two fully connected layers that further encode the BERT's pooled output and send the output to a classification layer. These hidden layers learn dataset-specific attributes; however, only a slight increase is observed. Model B3 processes the BERT's final hidden states corresponding to each token using a BiLSTM model. The striking performance gain achieved by B3 indicates that the use of the "[CLS]" token as the aggregate sequence representation may not duly retain vital information. Model B4 is similar to B3, except an attention layer is appended to the BiLSTM as mentioned in subsection 2.3.6, and it shows a relative improvement.

Model	Customized BERT Models	Precision	Recall	F1
B1	Default BERT	0.849	0.645	0.733
B2	BERT + nonlinear layers	0.779	0.737	0.758
B3	BERT + BiLSTM	0.731	0.795	0.762
B4	BERT + BiLSTM + Attention	0.786	0.747	0.766

Table 8: Performance metrics for customized BERT models on the PMI dataset.

3.4 Performance Comparison with Customized BERT Models with BioBERT Weights

Moreover, we test the performances of the customized BERT models by loading pre-trained weight of BioBERT v1.1, proposed by Lee et al. [Lee et al. 2020]. BioBERT is pre-trained on biomedical corpora such as PubMed abstracts and PMC full-text articles.

The results of customized BERT models initialized with BioBERT weights are reported in Table 9. Although BioBERT models show some slight performance improvements; overall, they are insignificant. BioBERT model is fine-tuned on BERT using biomedical texts but keeps the vocabulary consistent with the original. As a result, the vocabulary includes a lack of biomedical terminologies. Furthermore, due to the distinction between vernaculars and scholarly articles, the vocabulary may not capture meanings of social media texts effectively. The reasons behind the performance disparities among the BioBERT model variants can be induced from the BERT model variants, as both share the same architectures except for the weights.

Model	BERT Models with BioBERT Weights	Precision	Recall	F1
BB1	Default BERT	0.800	0.686	0.738
BB2	BERT + nonlinear layers	0.721	0.813	0.764
BB3	BERT + BiLSTM	0.759	0.760	0.760
BB4	BERT + BiLSTM + Attention	0.780	0.759	0.769

Table 9: Performance metrics for customized BERT models with BioBERT weights on the PMI dataset.

4 Conclusion

In this study, we designed a model to identify social media posts (i.e., tweets) with personal medication intake mentions by addressing the limitations of the LSTMs which are based on conventional word embedding. Due to a large number of out-of-vocabulary (OOV) tokens present in tweets, the use of word embedding alone to represent a token becomes ineffective. Hence, a character-level embedding that learns a representation for a token based on its character sequence is supplemented along with its word embedding. Furthermore, embedding-level attention is imposed to discover useful features from both embeddings without expanding dimensionality. The hub of our model, BiGRU, aptly learns a representation for a token from both previous and subsequent tokens in a sequence; however, the meaning of a token is dominated by adjacent tokens than faraway tokens due to the vanishing gradient problem. Another downside of BiGRU is that it fails to identify the tokens which highly influence the classification outcome. These drawbacks were resolved by capturing local context features using CNNs and assigning weighted precedence to tokens using context-aware attention. Our model achieves an F1-score of 0.772, which outperforms the vanilla BiGRU model and several customized BERT models to a considerable degree. In the future, we are planning to implement a model using other newly introduced derivatives of Transformer networks trained on the biomedical text.

References

- [Benton et al. 2012] Benton, A., Holmes, J. H., Hill, S., Chung, A., Ungar, L.: “medpie: an information extraction package for medical message board posts”; *Bioinformatics*, 28, 5 (2012), 743-744.
- [Chollet et al. 2015] Chollet, F. et al.: “Keras”; (2015) <https://github.com/fchollet/keras>.
- [Ding et al. 2018] Ding, P., Zhou, X., Zhang, X., Wang, J., Lei, Z.: “An Attentive Neural Sequence Labeling Model for Adverse Drug Reactions Mentions Extraction”; *IEEE Access*, 6, 1 (2018), 73305–73315.
- [Ellendorff et al. 2019] Ellendorff, T., Furrer, L., Colic, N., Aepli, N., Rinaldi, F.: “Approaching SMM4H with Merged Models and Multi-task Learning”; *Proceedings of the Fourth Social Media Mining for Health Applications (SMM4H) Workshop Shared Task*, Florence, Italy (August 2019), 58-61.
- [Gondane 2019] Gondane, S.: “Neural Network to Identify Personal Health Experience Mention in Tweets Using BioBERT Embeddings”; *Proceedings of the Fourth Social Media Mining for Health Applications (SMM4H) Workshop Shared Task*, Florence, Italy (August 2019), 110-113.
- [Goodfellow et al. 2016] Goodfellow, I., Bengio, Y., Courville, A.: “Deep Learning”; The MIT Press (2016).
- [Jiang et al. 2016] Jiang, K., Calix, R., Gupta, M.: “Construction of a Personal Experience Tweet Corpus for Health Surveillance”; *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, Berlin, Germany (August 2016), 128-135.
- [Jiang et al. 2017] Jiang, K., Tang, Y., Cook, G. E., Madden, M. M.: “Discovering Potential Effects of Dietary Supplements from Twitter Data”; *Proc. 2017 Int. Conf. Digital Health*, ACM, New York (July 2017), 119-126.
- [Jiang et al. 2018] Jiang, K., Feng, S., Song, Q., Calix, R. A., Gupta, M., Bernard, G. R.: “Identifying tweets of personal health experience through word embedding and LSTM neural network”; *BMC Bioinformatics*, 19, 8 (2018), 210.

- [Karisani and Agichtein 2018] Karisani, P., Agichtein, E.: “Did You Really Just Have a Heart Attack? Towards Robust Detection of Personal Health Mentions in Social Media”, CoRR, abs/1802.09130 (2018), 137-146.
- [Kiritchenko et al. 2018] Kiritchenko, S., Mohammad, S. M., Morin, J., de Bruijn B.: “NRC-Canada at SMM4H Shared Task: Classifying Tweets Mentioning Adverse Drug Reactions and Medication Intake”; Proceedings of the Social Media Mining for Health Applications Workshop at AMIA-2017, Washington, DC, (November 2017), 1-11.
- [Lee et al. 2020] Lee, J., Yoon, W., Kim S., Kim, D., Kim, S., So, C.H., Jang K.: “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”; *Bioinformatics*, 36, 4 (2020), 1234-1240.
- [Liu and Guo 2019] Liu, G., Guo, J.: “Bidirectional LSTM with attention mechanism and convolutional layer for text classification”; *Neurocomputing*, 337, 1 (2019), 325-338.
- [Mahata et al. 2018] Mahata, D., Friedrichs, J., Hitkul, Shah, R.R.: “pharmacovigilance - Exploring Deep Learning Techniques for Identifying Mentions of Medication Intake from Twitter”; arXiv (2018) <https://doi.org/10.48550/arXiv.1805.06375>.
- [Narui et al. 2020] Narui, H., Shu, R., Gonzalez-navarro, F. F., Ermon, S.: “Precision Health and Medicine”; Springer International Publishing, 843, 1 (2020).
- [Paszke et al. 2019] Paszke et al. A.: “PyTorch: An imperative style, high-performance deep learning library”; *Advances in Neural Information Processing Systems*, arXiv(2019) <https://doi.org/10.48550/arXiv.1912.01703>.
- [Pennington et al. 2014] Pennington, J., Socher, R., Manning, D.: “Global Vectors for Word Representations”; *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, ACL, Doha, Qatar (October 2014), 1532-1543.
- [Sarker et al. 2016] Sarker, A., O'Connor, K., Ginn, K., Scotch, M., Smith, K., Malone, D., Gonzalez, G.: “Social Media Mining for Toxicovigilance: Automatic Monitoring of Prescription Medication Abuse from Twitter”; *Drug Saf.*, 39, 3 (2016), 231-240.
- [Sarker and Gonzalez 2015] Sarker, A., Gonzalez, G.: “Portable Automatic Text Classification for Adverse Drug Reaction Detection via Multi-corpus Training”; *Biomed. Informatics*, 53, 1 (2015), 196-207.
- [Sarker and Gonzalez 2017] Sarker, A., Gonzalez-Hernandez, G.: “Overview of the second social media mining for health (SMM4H) shared tasks at AMIA 2017”; *CEUR Workshop Proc.*, 1996, 1 (2017), 43-48 <https://ceur-ws.org/Vol-1996/paper8.pdf>.
- [Vaswani et al. 2017] Vaswani, A. et al.: “Attention is all you need”; *Proc. 31st Int. Conf. Neural Information Processing Systems (NIPS’17)* (December 2017), 6000-6010.
- [Weissenbacher et al. 2018] Weissenbacher, D., Sarker, A., Paul, M., Gonzalez-Hernandez, G.: “Overview of the Third Social Media Mining for Health (SMM4H) Shared Tasks at EMNLP 2018”; *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop Shared Task*, Brussels, Belgium (October 2018), 13-16.
- [Weissenbacher et al. 2019] Weissenbacher, D. et al.: “Overview of the Fourth Social Media Mining for Health (SMM4H) Shared Tasks at ACL 2019”; *Proceedings of the Fourth Social Media Mining for Health Applications (SMM4H) Workshop Shared Task*, Florence, Italy (August 2019), 21-30.
- [Xherija 2018] Xherija, O.: “Classification of medication-related tweets using stacked bidirectional LSTMs with context-aware attention”; *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop Shared Task*, Brussels, Belgium (October 2018), 38-42.
- [Zhu and Jiang 2021] Zhu, M., Jiang, K.: “Semi-Supervised Language Models for Identification of Personal Health Experiential from Twitter Data: A Case for Medication Effects”; *Proceedings of the 20th Workshop on Biomedical Language Processing* (June 2021), 228-237.