Neural Computing and Applications (2023) 35:18161–18169
https://doi.org/10.1007/s00521-021-06614-2

S.I. : LATINX IN AI RESEARCH

Using weak supervision to generate training datasets from social media data: a proof of concept to identify drug mentions

Ramya Tekumalla[1] · Juan M. Banda[1]

Received: 3 January 2021 / Accepted: 4 October 2021 / Published online: 29 October 2021

Abstract
Twitter has been a remarkable resource for research in pharmacovigilance in the last decade. Traditionally, rule- or lexicon-based methods have been utilized for automatically extracting drug tweets for human annotation. The process of human annotation to create labeled sets for machine learning models is laborious, time consuming and not scalable. In this work, we demonstrate the feasibility of applying weak supervision (noisy labeling) to select drug data, and build machine learning models using large amounts of noisy labeled data instead of limited gold standard labelled sets. Our results demonstrate the models built with large amounts of noisy data achieve similar performance than models trained on limited gold standard datasets, hence demonstrating that weak supervision helps reduce the need to rely on manual annotation, allowing more data to be easily labeled and useful for downstream machine learning applications, in this case drug mention identification.

Keywords Weak supervision · Noisy learning · Pharmacovigilance · Twitter

1 Introduction

Social media, especially Twitter, has an abundance of data generated every day. On average, Twitter users generate 500 million tweets every day. In a review study [1] on Pharmacovigilance and Social media, 32 studies either used a lexicon-based method or a supervised learning method. Supervised learning techniques have achieved great success when there is strong supervision information like a large amount of training samples with ground-truth labels [2]. However, generating large training data with ground-truth labels is very expensive, tedious and time consuming. Since the availability of annotated data is limited, several researchers train the model with limited data and test the model on a small test set. These models are not scalable because they don't work well with large data. The new state-of-the-art models like BERT [3], trained on millions of parameters, demonstrated exceptional performance on several Natural Language Processing (NLP) tasks, while the supervised learning techniques are limited to smaller datasets due to manual annotation process.

While Twitter is an immense resource for research, on the downside, researchers cannot share tweet text directly and are only allowed to share tweet identifiers. Twitter users often delete their tweets, causing loss of data and wasted annotation effort (if the original authors do not save a copy). A research study [4], which utilized a publicly available annotated dataset [5], could hydrate only 66% of the original data and could not reproduce the original model to compare results. Thus, there is a huge need to avoid reliance on manual annotation of small datasets and to move to automatic annotation on large datasets.

Weak Supervision utilizes noisy, limited, or imprecise sources to provide a supervision signal for labeling large amounts of training data in a supervised learning setting [6]. To decrease the labelling costs, researchers have been using weaker forms of supervision, such as heuristically generating training data with external knowledge bases, patterns/rules, or other classifiers. The assumption behind our work is that the large volume of training data which can

✉ Juan M. Banda
jbanda@gsu.edu

Ramya Tekumalla
rtekumalla1@gsu.edu

[1] Department of Computer Science, Georgia State University, Atlanta, GA, USA

be collected using an automated labeling process can compensate for the inaccuracy in the labels. We base our assumption on the theory of noise-tolerant learning [7]. By imposing a bound on the labeling error and by using a sufficient number of training samples, models trained from very large data sets with noisy labels can be as good as those trained from data sets with clean labels. If successful, the use of such noise-tolerant learning can be scaled to several domains. Weak supervision is highly reliant on heuristics or labelling functions. Inclusion of bad heuristics will result in inclusion of poor data which in turn affects the machine learning models.

1.1 Theory of noisy learning

It is mathematically proven that addition of noise during the training of a neural network model has a regularization effect and, in turn, improves the robustness of the model [8]. However, the important question to envisage is how much noisy data are required to obtain a model with satisfactory performance. Simon [9] and later Aslam et al. [10] formulated the theory as a sample complexity bound which is also verified in a phenotype research [11]. The bound can be calculated as:

$$m \geq O\left[\frac{VC(H)}{\gamma(1-2\tau)^2} + \frac{\mathbf{log}(1/\boldsymbol{\delta})}{\gamma(1-2\tau)^2}\right]$$

where $\gamma > 0$ and $0 \leq \delta \leq 1$. where τ as the random classification error for data distribution of observations and noisy labels, H as the class of learning algorithms to which our models belong, $\hat{h}$ as a model in H and trained on a set of m observations drawn from data, h* as a model in H that best fits the target distribution consisting of correct observations and correct labels, $\varepsilon(\hat{h})$ as the generalization error of $\hat{h}$ and $\varepsilon(h^*)$ as the generalization error of h*.

The case $\tau = 0$ corresponds to observation data with clean labels and the case $\tau = 0.5$ represents the random flipping of labels that makes learning impossible. For a given error bound γ, probability $1 - \delta$, and classification error rate τ, a learning algorithm can learn equally well from approximately $\mathbf{m^*(1-2\tau)^2}$ observations of noisy data of what it can learn from m observations of clean data. The important aspect to note is that it is easier to obtain $\mathbf{m^*(1-2\tau)^2}$ noisy observations than to acquire m clean data. We computed the theoretical bounds, and the results are presented in the Results section.

2 Related work

Pharmacovigilance is defined as the science and activities relating to the detection, assessment, understanding and prevention of adverse effects or any other possible drug-related problems [12]. Machine learning and NLP techniques have been applied to several applications in pharmacovigilance such as identifying adverse drug reactions [13], analyzing twitter data for post marketing surveillance [14], analyzing prescription drug abuse, applications to identify misspelled drug mention tweets in tweets to obtain more data [15, 16]. Several classical and deep learning models were utilized for supervised learning. While the supervised learning approach is efficient, there is a bottleneck to using the approach due to the manual annotation process involved.

In this work, we demonstrate the usage of weak supervision to identify drug mentions from noisy Twitter data. Instead of manually annotating data, we utilized a heuristic approach to curate the noisy data, i.e., silver standard data to train several machine learning models. We emphasize that we did not add noise nor modify the data. Instead of using gold standard annotated labels, we made use of our noisy data and trained several machine learning models to identify drug tweets.

3 Data preparation

In our previous work [17], we mined over 6 million English drug tweets from 9 billion tweets utilizing a heuristic approach using Social Media Mining Toolkit (SMMT) [18]. The 6 million tweets contain considerable noise due to several irrelevant terms (Example: solution, predator) in the dictionary. In order to eliminate irrelevant noise, we cleaned up the drug dictionary by removing terms which are commonly used in English (Example: patch). Additionally, we also removed all the terms with length greater than 38 characters since the longest term tagged from 9 billion tweets was only 37 characters. We used the updated drug dictionary which consists of 19,643 terms and separated 4,214,737 tweets from 6 million tweets. Each tweet text consists of at least one drug term from the dictionary. All the tweets were preprocessed by removing emojis, emoticons, URLs, leading and trailing whitespaces. All the preprocessed drug tweets were labelled as "drug tweets." Listed below are a few samples of the preprocessed tweets obtained through heuristics. The drug terms are highlighted in bold.

(1) "health **melatonin** and exercise key combination for helping with alzheimers"

(2) “i hate having breathing problems to where i have to take up to 2–3 **xanax** at once just to slow down my heart beat”

(3) “hopefully this **tylenol** breaks my fever”

Tweet text cannot be shared publicly, and hence tweet ids are made available through our dataset paper [17] which can be hydrated using SMMT. We collected an equal number of non-drug tweets from Internet Archive [19] which is the same source utilized in our previous work [17] in order to have a language balanced dataset. To avoid the language bias, we retrieved non drug tweets randomly from the same years as the drug tweets. All the non-drug tweets were preprocessed and labeled as “non-drug tweets.” We created 10 different class balanced training sets for 7 different training sizes. A total of 70 training sets were used in our methodology. Table 1 describes the details of each training set and the total number of datasets. To create the training sets, the data were randomly sampled for all the training sizes from the pool of drug and non-drug tweets. The smaller training sizes (e.g., 100,000, 200,000, 300,000) have no overlapping data among different training sets. However, for the larger training sizes (e.g., 2 million), there is an overlap in the data since the drug tweets pool contains only ~ 4 million tweets. Each training set in a training size contains an equal number of drug and non-drug tweets. For example, the 100 k training set consists of randomly shuffled 100,000 drug tweets and 100,000 non-drug tweets. For each training set in the training size, we split the data into 80% (training dataset) and 20% (validation dataset).

In order to create the test set, we collected publicly available manually and expertly curated datasets [5, 20]. Due to Twitter’s terms and conditions, the authors could only release the tweet identifiers and we hydrated the tweet json objects and extracted tweet text. A total of 7215 annotated tweets were publicly available at the time of experiments and were hydrated and preprocessed using SMMT. To have a balanced test set, we added 7215 non-drug tweets totaling 14,430 tweets, which is used as the test set in our methods. We would like to emphasize that we did not manually annotate any tweets and instead used publicly available, manually and expertly annotated drug tweets in our test set.

4 Methods

We experimented with several classical and deep learning models using our preprocessed training and test datasets.

4.1 Classical models

We experimented with five classifiers: Naïve Bayes, Logistic Regression (LR), Support Vector Machines (SVM), Random Forest and Decision Trees using the scikit-learn [21]. Support Vector Machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. We used a LinearSVC which is similar to SVC, but implemented using liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples. Naive Bayes (NB) methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable. We used the Multinomial Naive Bayes which implements the naive Bayes algorithm for multinomial distributed data and is one of the two classic naive Bayes variants used in text classification. A Random Forest (RF) is a meta-estimator that fits a number of decision tree classifiers on various subsamples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The Decision Tree (DT) Classifier uses a CART algorithm (Classification And Regression Tree). CART is a nonparametric decision tree learning technique that produces either classification or regression trees, depending on whether the dependent

Table 1 Training sets and description

Training size	Description	No. of training sets used
100 k	100,000 drug tweets + 100,000 non drug tweets	10
200 k	200,000 drug tweets + 200,000 non drug tweets	10
300 k	300,000 drug tweets + 300,000 non-drug tweets	10
500 k	500,000 drug tweets + 500,000 non-drug tweets	10
1 M	1,000,000 drug tweets + 1,000,000 non-drug tweets	10
2 M	2,000,000 drug tweets + 2,000,000 non-drug tweets	10
3 M	3,000,000 drug tweets + 3,000,000 non-drug tweets	10

variable is categorical or numeric, respectively. However, the scikit-learn library uses an optimized version of the CART which does not support categorical values. For all the models, scikit learn's TF-IDF vectorizer was used to convert raw tweet text to TF-IDF features and return the document-term matrix which is sent to the model.

4.2 Deep learning models

We experimented with 5 different deep learning models which include Transformers, CNN and LSTM models. Bidirectional Encoder Representations from Transformers (BERT) [3] has been the state-of-the-art language representation model for solving a wide range of tasks, such as question answering, language inference, sentence prediction and text classification. We experimented with the "bert-large-uncased" model, which is of 24-layer, 1024-hidden, 16-heads, 340 M parameters and trained on lower-cased English Wikipedia text and book corpus [22]. Bidirectional Encoder Representations from Transformers for Biomedical Text Mining (BioBERT) [23] is a domain-specific language representation model pre-trained on large-scale biomedical corpora. The BioBERT model architecture used for our experiment is 12-layer, 768 hidden size, 12-heads, 1 M parameters and trained on PubMed baseline abstracts. The final architecture used in Transformers is Robustly Optimized BERT Pretraining Approach (RoBERTa) [24] which has an improved pre-training procedure over BERT. We used the "roberta-large" model which is of 24-layer, 1024-hidden, 16-heads, 355 M parameters RoBERTa using the BERT-large architecture. For implementation, we utilized Simple Transformers [25] which seamlessly worked with the Natural Language Understanding (NLU) architectures made available by Hugging Face's Transformers models [26].

Although the Convolutional Neural Networks (CNN) were originally built for image processing, they proved to achieve exceptional results when applied to text data as [27–29]. We incorporated a basic CNN architecture with convolutional layers, embedding layer, dropout layers, maxpooling layers, dense layers and a flatten layer. We used the Keras implementation of CNN model by Text Classification Algorithms: A survey [30]. For the experiment, we used Adam Optimizer, Relu Activation function and RedMed [31] Embedding model. All the CNN experiments were run on 10 epochs and 2048 batch size.

The final model used in our experiments was Bidirectional LSTM [32] which belongs to a larger category of Neural Networks called Recurrent Neural Networks (RNN). In RNN, the neural net considers the information of previous nodes in a very sophisticated method which allows for better semantic analysis of the structures in the dataset and therefore is a powerful technique for text classification tasks. The LSTM architecture consists of three different gates: Input, Output and Forget gate which operate together to decide what information to remember and what to forget in the LSTM cell over an arbitrary time. We incorporated a basic LSTM architecture with an Embedding layer, Bidirectional LSTM, dropout layer and dense layer. We used the Keras implementation of CNN model by Text Classification Algorithms: A survey [30]. We used the RedMed [31] Embedding model with the following hyper-parameters for the experiment: Adam Optimizer, bidirectional set to true, max sequence length 280, dropout 0.2 and softmax activation function were used in our experiment. All the experiments were run on 10 epochs with batch size 1024.

In order to select the best word embedding model, we experimented with 6 different word embeddings listed in Table 2 and adopted the best embedding model based on ROC curves from Fig. 1. The best results were obtained when the RedMed embedding model was used. The RedMed model is trained on a corpus of more than 500 million comments over 2500 subreddits. The model has an embedding size of 64 dimensions, a window size of 7 and a minimum count of 5.

5 Experimental results

In order to evaluate the results, we used the following metrics: precision (P), recall (R), F-measure (F) and accuracy (A). Table 3 presents the results of the SVM model on all training sets for the 100 k training size. Training set number 9 has the best F-measure score when compared with all the other training sets. For example, for the 100 k training size, in classical models, a total of 50 experiments were performed (5 models * 10 training sets * 1 training size). Out of 50 experiments, SVM model training set number 9 obtained the best results. The experiment with the best F measure is used as the metric while plotting Fig. 4. A total of 700 experiments (10 models * 10 different training sets * 7 different training sizes) were implemented as part of our experiment.

Table 4 presents the best F-measure scores in each training size in all the classical models. Figures 2, 3 and 4 depict the precision, recall and F-measure for the classical models. In classical models, we could obtain a maximum F-Measure of **0.9092** for the 300,000 drug samples dataset.

Although the SVM model performed the best amongst all the models in all training sizes, the important observation to note here is that the Logistic Regression and Naive Bayes models came close to SVM in performance.

For the deep learning models, CNN and LSTM, we obtained the probability scores for the test set and compiled

Table 2 Details of Embeddings

#	Embedding name and source	Details
em1	Drug Chatter Twitter [16]	1B drug tweets from user timelines; window size 5 and dimension 400
em2	Glove [33]	840B tokens, 2.2 M vocab, cased, 300d vectors
em3	Twitter Word2vec Embeddings [34]	400 million Twitter tweets; Negative sampling; Skip-gram architecture; Window of 1; subsampling rate of 0.001; Vector size of 400
em4	glove.twitter.27B [33]	2B tweets, 27B tokens, 1.2 M vocab, uncased,200d vectors
em5	RedMed Model [31]	3 M tokens, 64d; Reddit drug posts
em6	Glove [33]	42B tokens, 1.9 M vocab, uncased, 300d vectors

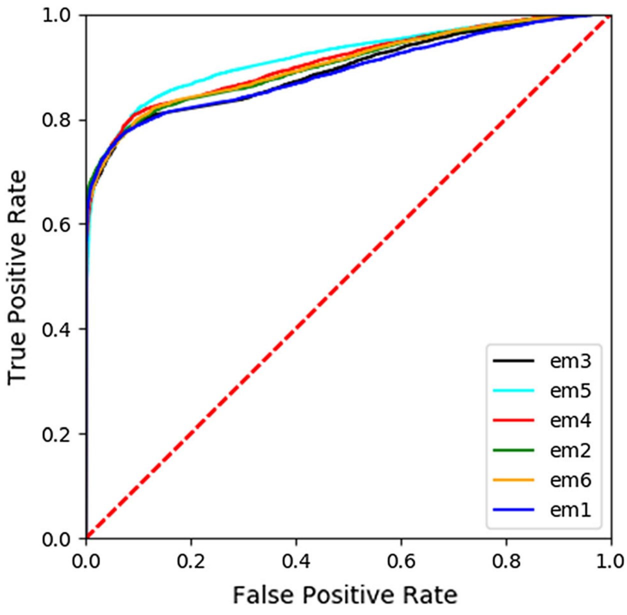


Fig. 1 ROC curves for all embedding models

Table 3 SVM model results for all training sets for the training size 100 k

Training set #	P	R	F	A
1	0.9943	0.8012	0.8874	0.8983
2	0.9946	0.8004	0.887	0.8981
3	0.9946	0.8028	0.8884	0.8992
4	0.9939	0.7990	0.8859	0.8971
5	0.9938	0.7869	0.8784	0.8910
6	0.9938	0.8028	0.8881	0.8989
7	0.9941	0.7955	0.8838	0.8954
8	0.9939	0.8037	0.8888	0.8994
9	**0.9933**	**0.8069**	**0.8905**	**0.9008**
10	0.9941	0.8018	0.8876	0.8985

Table 4 F-measure of best classical models

Training Size	LR	SVM	NB	RF	DT
100 k	0.8429	**0.8905**	0.8462	0.771	0.7512
200 k	0.8597	**0.8968**	0.8626	0.7821	0.7194
300 k	0.8708	**0.9092**	0.8749	0.8028	0.7582
500 k	0.8784	**0.9023**	0.8861	0.8116	0.7465
1 M	0.8907	**0.9058**	0.8906	0.8231	0.775
2 M	0.8948	**0.9048**	0.8961	0.8409	0.7592
3 M	0.8984	**0.9049**	0.8989	0.8464	0.7262

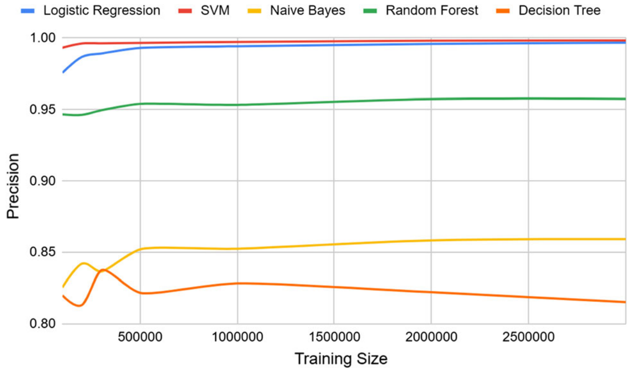


Fig. 2 Precision for all best classical models

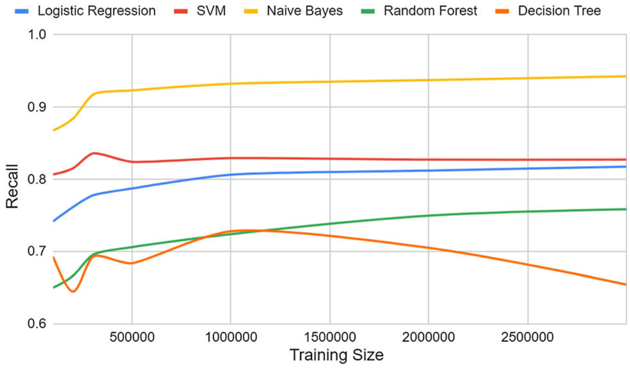


Fig. 3 Recall for all best classical models

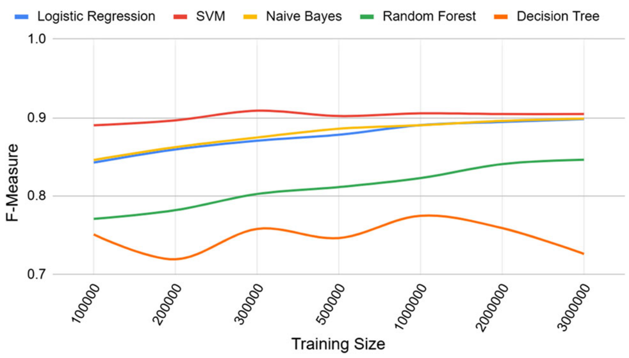
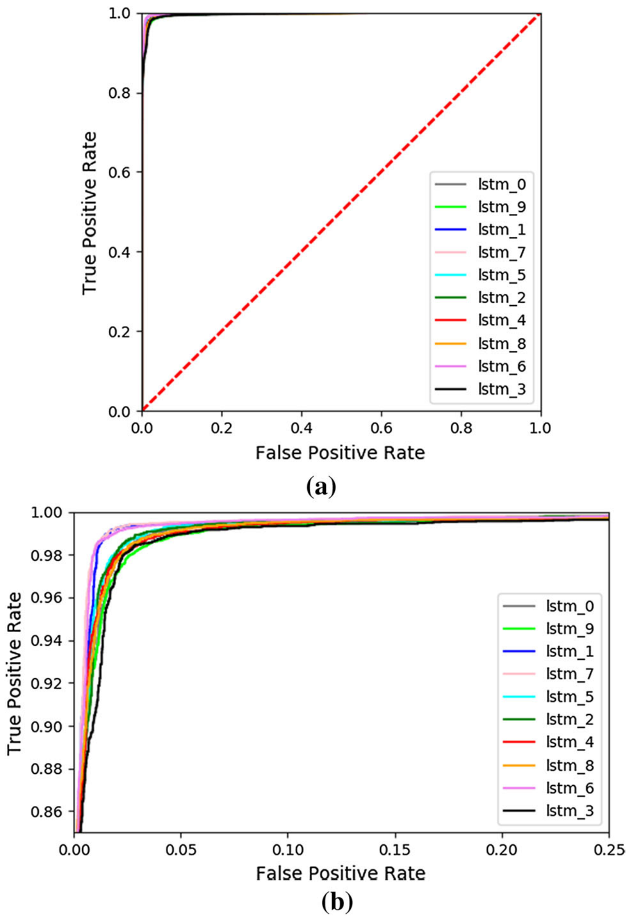


Fig. 4 F-measure for all best classical models



(a)

(b)

Fig. 5 **a** ROC curves for all the 10 training sets for the 100 k training size. **b** Enlarged version of (**a**)

the ROC curves and determined the best training set in a training size. Further, the best model ROC curve was used to determine the cutoff threshold. For each training size, we set a cutoff threshold using ROC curves. Figure 5a depicts the ROC curves of the LSTM model for the training size 100 k, and Fig. 5b depicts the enlarged version of the 5a to which offers better insight to 5a.

Based on the ROC curves, training set 7 was determined to be the best model in the 100 k training size. For the training size 100 k, based on ROC curves (Fig. 5), the cutoff threshold was determined to be 0.65. All the tweets with probability scores greater than 0.65 were classified as drug tweets, and the other tweets are classified as non-drug tweets. Table 5 presents the best F-measure scores of deep learning models for all the best training sets in all the training sizes. Figures 6, 7 and 8 depict the precision, recall and F-measure for the deep learning models. In deep learning models, we could obtain a maximum of 0.9951 F-measure score for 100,000 drug samples.

The CNN and LSTM models when used with RedMed embeddings performed better when compared with other embedding models (e.g., Glove models which are trained on 840B tokens and 2.2 Million vocab). However, surprisingly the transformer models (BERT) outperformed the results of CNN and LSTM and BioBERT (biomedical representation model). This might be due to Twitter being a non-medical social media platform where English vocabulary is predominant than medical vocabulary. The primary objective of this research is to demonstrate our approach of using noisy labels instead of a gold standard annotated datasets in the context of social media data mining, but not to identify the best machine learning model which works for weak supervision. The results from Tables 4 and 5 validate our hypothesis.

5.1 Experiments with gold standard dataset

In order to validate our results with the gold standard dataset, we performed experiments only with the gold standard dataset. We used a 75–25 split of gold standard dataset as train and test data and trained both classical and deep learning models. The results obtained from the experiments are presented in Tables 6 and 7.

The SVM model obtained an F-measure of 0.9892 and outperformed all the other models in classical models. Unsurprisingly, the BERT model's performance was

Table 5 F-measure of best Deep Learning Models

# Size	BERT	Bio BERT	RoBERTa	CNN	LSTM
100 k	**0.9951**	0.9296	0.9339	0.8675	0.9869
200 k	0.9282	0.9287	0.9333	0.8566	**0.9625**
300 k	0.9324	0.9299	0.9371	0.8558	**0.9442**
500 k	0.9270	0.9264	**0.9655**	0.8549	0.9389
1 M	0.9705	0.9257	**0.9891**	0.8392	0.9312
2 M	0.9231	0.9152	**0.9729**	0.8118	0.9152
3 M	**0.9719**	0.9125	0.9517	0.8200	0.9180

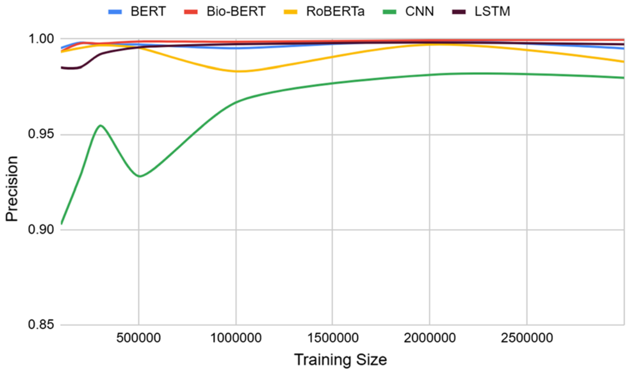


Fig. 6 Precision of all best deep learning models

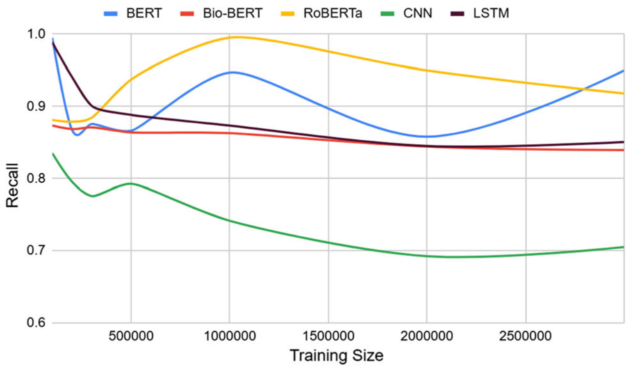


Fig. 7 Recall of all best deep learning models

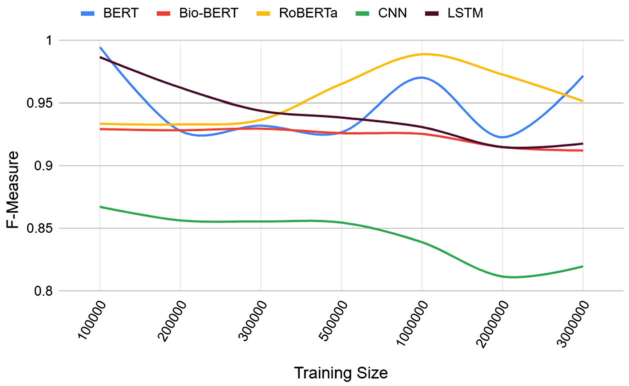


Fig. 8 F-measure of all best deep learning models

superior when compared to classical models and all deep learning models.

5.2 Calculating theoretical bounds

Based on the theory of noisy learning, we calculated the minimum number of noisy samples using the accuracy of the best and worst performing models. The following table presents the number of noisy observations required when there are 10,822 samples (75% of the gold standard data) available. The minimum number of noisy observations required when $\gamma = 0.05$ and $\delta = 0.05$ are presented in Table 8.

Table 6 Results of classical models when trained on gold standard dataset

Classifier	P	R	F	A
LR	0.9635	0.9494	0.9564	0.9573
SVM	**0.9943**	**0.9842**	**0.9892**	**0.9894**
NB	0.7191	0.9898	0.8330	0.8043
RF	0.8032	0.7685	0.7855	0.7929
DT	0.9447	0.9601	0.9523	0.9526

Table 7 Results of deep learning models when trained on gold standard dataset

Classifier	P	R	F	A
CNN	0.9146	0.8737	0.8937	0.8968
RNN	0.9855	0.9882	0.9868	0.9869
BERT	**0.9978**	**0.9978**	**0.9978**	**0.9977**
BioBERT	0.9850	0.9973	0.9911	0.9908
RoBERTa	0.9967	0.9978	0.9973	0.9972

According to Table 8, for a generalization error of 0.05, we would require 42,021 minimum noisy samples instead of 10,822 gold standard samples. Theoretically, we require a minimum 42,021 noisy samples.

Table 9 presents the comparison of results between the best performing models when trained on gold and silver standard data. When trained on gold standard data, BERT model obtained the best F-measure score (0.9978) when trained on 10,822 samples (75% of the gold standard data). However, the BERT model obtained 0.9951 F-measure score when trained on 100,000 samples of silver standard data. The important observation here is that it is relatively easier to obtain 100,000 samples of silver standard data than to obtain 10,000 samples of gold standard data.

Table 8 Number of minimum noisy samples required

Accuracy	Minimum number of noisy samples	Model
0.9978	14,558	BERT
0.7930	42,021	DT

Table 9 Comparison of gold standard vs silver standard results

Data	No. of drug samples	P	R	F	A	Best model
Gold standard	7215	0.9978	0.9978	0.9978	0.9977	BERT
Silver standard	100,000	0.9953	0.9950	0.9951	0.9951	BERT
Silver standard	100,000	0.9852	0.9410	0.9626	0.9634	LSTM
Silver standard	300,000	0.9919	0.9009	0.9442	0.9468	LSTM
Silver standard	500,000	0.9954	0.9374	0.9655	0.9665	RoBERTa
Silver standard	1,000,000	0.9832	0.9953	0.9892	0.9891	RoBERTa
Silver standard	2,000,000	0.9971	0.9500	0.9730	0.9736	RoBERTa
Silver standard	3,000,000	0.9951	0.9500	0.9720	0.9726	BERT

In this work, our primary contribution is to demonstrate the feasibility of using weak supervision to identify drug mentions from noisy Twitter data. We utilized silver standard dataset instead of gold standard dataset to train the machine learning models as curating gold standard dataset is tedious and laborious. The primary motivation behind this work is to demonstrate the results of models when trained on noisy data on several training sizes. Though the results demonstrated that with the increase in training size there is an increase in the performance of the model, we wanted to emphasize with proof that noisy data can be utilized for training sizes as small as 100,000 samples and as large as 3,000,000 samples. While seemingly a straightforward task, we would like to emphasize that this is the first application to utilize weak supervision in the field of pharmacovigilance.

6 Future work

There are several directions in which this work can progress. Firstly, the training set used in our experiments was compiled by utilizing a heuristic dictionary based approach. This approach does not acquire tweets with misspellings which contain potentially important data. On twitter COVID-19 drug mentions research [15], considering misspellings yielded 20 percent additional data. Thus, in future work, we would employ misspelling modules to obtain more training data. Secondly, we artificially created a balanced training dataset, but in reality, drug tweets comprise less than 1% of all the tweets generated. We would like to perform several experiments with trade-off between classes and balance. Finally, instead of utilizing a lexicon, we would like to employ labelling functions to annotate a tweet.

7 Conclusion

Supervised learning techniques are successful when large annotated datasets are available. However, labeling data has become a bottleneck due to cost associated with it. In this work, we demonstrate the feasibility of utilizing weak supervision to obtain results similar to supervised learning. This approach can help reduce the need for manual annotation which saves time and resources. Further, the models from Weak Supervision are scalable and can be utilized with large data. Additionally, the approach can easily be extended to several other applications by generating heuristics and curating silver standard datasets.

Acknowledgements We would like to thank Stephen Fleischman and HP labs for providing us with a server with 8 GPUs to perform our experiments during our research server downtime.

Author contributions RT and JMB were involved in conceptualization, data curation, formal analysis, methodology and writing—review & editing.

Availability of data and material The dataset used for training the models is available via Zenodo [17]. The trained models are available on request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Sarker A, Ginn R, Nikfarjam A et al (2015) Utilizing social media data for pharmacovigilance: A review. J Biomed Inform 54:202–212. https://doi.org/10.1016/j.jbi.2015.02.004
2. Zhou Z-H (2017) A brief introduction to weakly supervised learning. Natl Sci Rev 5:44–53. https://doi.org/10.1093/nsr/nwx106
3. Devlin J, Chang M-W, Lee K, Toutanova K (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv [cs.CL]

4. Cocos A, Fiks AG, Masino AJ (2017) Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in Twitter posts. J Am Med Inform Assoc 24:813–821. https://doi.org/10.1093/jamia/ocw180
5. Nikfarjam A, Sarker A, O'Connor K et al (2015) Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. J Am Med Inform Assoc 22:671–681. https://doi.org/10.1093/jamia/ocu041
6. Ratner A, Varma P, Hancock B, et al (2019) Weak Supervision: A New Programming Paradigm for Machine Learning. http://ai.stanford.edu/blog/weak-supervision/. Accessed 23 Nov 2020
7. Angluin D, Laird P (1988) Learning from noisy examples. Mach Learn 2:343–370. https://doi.org/10.1007/BF00116829
8. Bishop CM (1995) Training with noise is equivalent to Tikhonov regularization. Neural Comput 7:108–116. https://doi.org/10.1162/neco.1995.7.1.108
9. Simon HU (1996) General bounds on the number of examples needed for learning probabilistic concepts. J Comput System Sci 52:239–254. https://doi.org/10.1006/jcss.1996.0019
10. Aslam JA, Decatur SE (1996) On the sample complexity of noise-tolerant learning. Inf Process Lett 57:189–195. https://doi.org/10.1016/0020-0190(96)00006-3
11. Agarwal V, Podchiyska T, Banda JM et al (2016) Learning statistical models of phenotypes using noisy labeled training data. J Am Med Inform Assoc 23:1166–1173. https://doi.org/10.1093/jamia/ocw028
12. Lindquist M (2007) The need for definitions in pharmacovigilance. Drug Saf 30:825–830. https://doi.org/10.2165/00002018-200730100-00001
13. O'Connor K, Pimpalkhute P, Nikfarjam A et al (2014) Pharmacovigilance on twitter? Mining tweets for adverse drug reactions. AMIA Annu Symp Proc 2014:924–933
14. Pain J, Levacher J, Quinquenel A, Belz A (2016) Analysis of Twitter data for postmarketing surveillance in pharmacovigilance. In: Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT). pp 94–101
15. Tekumalla R, Banda JM (2020) Characterizing drug mentions in COVID-19 Twitter Chatter. In: Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020. Association for Computational Linguistics, Online, 2020. https://doi.org/10.18653/v1/2020.nlpcovid19-2.25
16. Sarker A, Gonzalez G (2017) A corpus for mining drug-related knowledge from Twitter chatter: Language models and their utilities. Data Brief 10:122–131. https://doi.org/10.1016/j.dib.2016.11.056
17. Tekumalla R, Asl JR, Banda JM (2020) Mining Archive. org's Twitter Stream Grab for Pharmacovigilance Research Gold. In: Proceedings of the International AAAI Conference on Web and Social Media. pp 909–917
18. Tekumalla R, Banda JM (2020) Social Media Mining Toolkit (SMMT). Genomics Inform 18:e16. https://doi.org/10.5808/GI.2020.18.2.e16
19. Machine W (2015) The Internet Archive. Searched for https://www.icannorg/icp/icp-1html
20. Klein A, Sarker A, Rouhizadeh M et al (2017) Detecting personal medication intake in Twitter: an annotated corpus and baseline classification system. BioNLP 2017:136–142
21. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: Machine Learning in Python. J Mach Learn Res 12:2825–2830
22. Zhu Y, Kiros R, Zemel R, et al (2015) Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In: Proceedings of the IEEE international conference on computer vision. pp 19–27
23. Lee J, Yoon W, Kim S et al (2020) BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 36:1234–1240. https://doi.org/10.1093/bioinformatics/btz682
24. Liu Y, Ott M, Goyal N, et al (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv [cs.CL]
25. Rajapakse T (2020) Simple Transformers Available at https://github.com/ThilinaRajapakse/simpletransformers
26. Wolf T, Debut L, Sanh V, et al (2019) HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv e-prints. arXiv:1910.03771
27. Hu B, Lu Z, Li H, Chen Q (2014) Convolutional Neural Network Architectures for Matching Natural Language Sentences. In: Ghahramani Z, Welling M, Cortes C, et al (eds) Advances in Neural Information Processing Systems 27. Curran Associates, Inc., pp 2042–2050
28. Dos Santos C, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp 69–78
29. Wang P, Xu J, Xu B, et al (2015) Semantic clustering and convolutional neural network for short text categorization. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). pp 352–357
30. Kowsari K, Jafari Meimandi K, Heidarysafa M et al (2019) Text Classification Algorithms: A Survey. Information 10:150. https://doi.org/10.3390/info10040150
31. Lavertu A, Altman RB (2019) RedMed: Extending drug lexicons for social media applications. J Biomed Inform 99:103307. https://doi.org/10.1016/j.jbi.2019.103307
32. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
33. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp 1532–1543
34. Godin F (2019) Improving and Interpreting Neural Networks for Word-Level Prediction Tasks in Natural Language Processing. PhD thesis, PhD Thesis, Ghent University, Belgium, 2019. 35