Training Data Augmentation for Detecting Adverse Drug Reactions in User-Generated Content

Mesbah, Sepideh; Yang, Jie; Sips, Robert-Jan; Torre, Manuel Valle; Lofi, Christoph; Bozzon, Alessandro; Houben, Geert-Jan

DOI
10.18653/v1/D19-1239

Publication date
2019

Document Version
Final published version

Published in
International conference on Empirical Methods in Natural Language Processing (EMNLP)

Citation (APA)
Mesbah, S., Yang, J., Sips, R-J., Torre, M. V., Lofi, C., Bozzon, A., & Houben, G-J. (2019). Training Data Augmentation for Detecting Adverse Drug Reactions in User-Generated Content. In *International conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 2349–2359) https://doi.org/10.18653/v1/D19-1239

Training Data Augmentation for Detecting Adverse Drug Reactions in User-Generated Content

Sepideh Mesbah[1], Jie Yang[2], Robert-Jan Sips[3],
Manuel Valle Torre[1], Christoph Lofi[1], Alessandro Bozzon[1], and Geert-Jan Houben[1]

[1]Delft University of Technology, Van Mourik Broekmanweg 62628 XE Delft , the Netherlands
[2]Amazon[*], 440 Terry Ave N, Seattle, WA 98109, USA
[3]MyTomorrows, Anthony Fokkerweg 61, 1059 CP Amsterdam, the Netherlands
s.mesbah@tudelft.nl, jiy@amazon.com, r.sips@mytomorrows.com
{m.valletorre, c.lofi, a.bozzon, g.j.p.m.houben}@tudelft.nl

Abstract

Social media provides a timely yet challenging data source for adverse drug reaction (ADR) detection. Existing dictionary-based, semi-supervised learning approaches are intrinsically limited by the coverage and maintainability of laymen health vocabularies. In this paper, we introduce a data augmentation approach that leverages variational autoencoders to learn high-quality data distributions from a large unlabeled dataset, and subsequently, to automatically generate a large labeled training set from a small set of labeled samples. This allows for efficient social-media ADR detection with low training and re-training costs to adapt to the changes and emergence of informal medical laymen terms. An extensive evaluation performed on Twitter and Reddit data shows that our approach matches the performance of fully-supervised approaches while requiring only 25% of training data.

1 Introduction

Adverse Drug Reactions (ADRs) is the fourth leading cause of death in the United States (Chee et al., 2011). ADR detection is, therefore, a critical element of drug safety. Studies have shown that clinical trials are not able to fully characterize drugs' adverse effects (Harpaz et al., 2012; Chee et al., 2011; Ahmad, 2003). Traditional techniques of post-market ADR mainly rely on voluntary and mandatory reporting of ADRs by patients and health providers, but they suffer from delays in reporting, under-reporting, or data incompleteness (Sarker et al., 2015).

Social media is becoming a preferred channel for millions of users and patients to share, discuss, and seek health information (He et al., 2017); such user-generated content can, therefore, provide valuable insights for monitoring Adverse Drug Reactions (ADRs) from an additional point of view (Lee et al., 2017; Sarker and Gonzalez, 2015; Aramaki et al., 2011). ADR detection from social media is, however challenging, as online users report ADRs using a different language style and terminology that largely depend on the user's medical proficiency, as well as the type of online medium (e.g., health forums vs micro-post social networks). In particular, laymen often use diverse dialects (Karisani and Agichtein, 2018) when describing medical concepts, and make abundant use of informal terminology.

Existing approaches for detecting terms in informal medical language mainly rely on semi-manually generated dictionaries (e.g. laymen health dictionaries) (Zeng and Tse, 2006), or supervised machine-learning-based sequence classifiers (Huynh et al., 2016; Chowdhury et al., 2018). Due to the language dynamicity in online and offline communication (Kershaw et al., 2016; Zanzotto and Pennacchiotti, 2012), there is a constant emergence of new informal medical terms. This results in a lack of coverage and maintainability of laymen health vocabularies. While showing superior performance, machine learning approaches often need to be trained for specific Web communities and platforms due to differences in the underlying language models; this results in high costs for manual annotation of training data, which for many domains is only sparsely available (Deriu et al., 2017).

More recently, researchers have started to investigate techniques for expanding the size of manually created training data. Often, sentence similarity implemented with embedding techniques (Mikolov et al., 2013; Le and Mikolov, 2014) is used to discover similar sentences, and then annotations are automatically propagated to those sentences (Mesbah et al., 2018). While these techniques have indeed shown to reduce the cost of

[*] Work performed before joining Amazon.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 2349–2359,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

training, they are typically limited by *availability of the existing data* as the reliability of annotation propagation suffers when sentences are not similar enough. Therefore, in this paper we focus on how to automatically *generate* high quality training data for Adverse Drug Reaction detection with minimal human supervision and costs.

Original Contribution. In this paper, we propose to generate artificial sentences closely mimicking existing training data; such artificial sentences are then annotated automatically via label propagation. This contrasts existing approaches expanding manually created training data set by discovering additional existing sentences in a dataset.

We build our method upon variational autoencoders (VAE), a deep probabilistic neural model which learns latent text features and their distributions effectively. In contrast to other approaches using variational autoencoders for text generation, we modify the mechanism for generating new artificial samples such that we obtain samples structurally and semantically similar to a specific subset of the original data. This allows us to generate sentences similar to those in the pre-existing human-labeled ADR training usable for classifier training set by taking advantage of the implicit semantics contained in the larger unlabeled dataset.

We evaluate the proposed method on a standard Twitter dataset and on a large new dataset for the Reddit platform we created with the help of expert annotators. The dataset is available on the companion Website (Companion, 2019). Results show that our approach achieves superior or comparable performance with significantly less training data (*reduced by 75%*) than state-of-the-art training methods.

2 Related Work

The terminology adopted in most social communities makes heavy use of slang or indirect descriptions, which is often lacking with respect to grammar and orthography; in addition, it is also constantly evolving and differs between communities. This makes the use of established techniques relying on expert-curated dictionaries (Leaman et al., 2010; Soldaini and Goharian, 2016) of consumer health vocabulary or fully-supervised machine learning-based classifiers (Huynh et al., 2016; Chowdhury et al., 2018) expensive, and in many cases even prohibits their use. While techniques to lessen the costs of training like distant supervision (Mintz et al., 2009) or bootstrapping (Tsai et al., 2013; Blum and Mitchell, 1998) can provide some support, their performance has been shown to be limited.

Some recent work has started to address the issues of size and cost of ADR training data (Lee et al., 2017; Cocos et al., 2017; Nikfarjam et al., 2015). Lee et al. (2017) explores different types of unlabeled data and a small training set to generate phrase embeddings, so to *classify* the tweets that indicate adverse drug event in a semi-supervised way. In contrast, our work focuses on detecting the *actual ADR span* in the text of the user generated posts rather than just classifying the whole post as containing an ADR mention. Nikfarjam et al. (2015) and Cocos et al. (2017) augment traditional supervised methods with additional features such as pre-trained word representation vectors, to improve performance and to be less dependent on large training sets. The resulting BLSTM-RNN (Cocos et al., 2017) technique, which achieves state-of-the-art performance, is also evaluated in our experiments (see Section 5.1). Rather than adding new features or proposing new ADR detector models, our work focuses on the generation of new labeled data samples from small annotated training sample using deep probabilistic models.

Different from the above approaches, embedding based methods (Le and Mikolov, 2014; Mikolov et al., 2013) learn vector representations of words or paragraphs to capture semantic relationships among words. Such methods are, therefore, useful to find sentences similar to the labeled training data, thereby expanding the size of the training data. Embedding based methods, however are limited by the existing sentences available in a given corpus. In contrast, our approach is capable of generating new sentences not existing in the corpus, thus largely expanding the training data. Our approach for generating additional labeled training data is inspired by (Bowman et al., 2016), where VAEs are used to learn a generative model of text for sentence generation. Bowman et al. (2016), however, only tackles the general problem of sentence generation. To the best of our knowledge, our work is the first that investigate VAEs as a tool for training data expansion, so as to enhance machine learning performance with limited amount of labeled data.

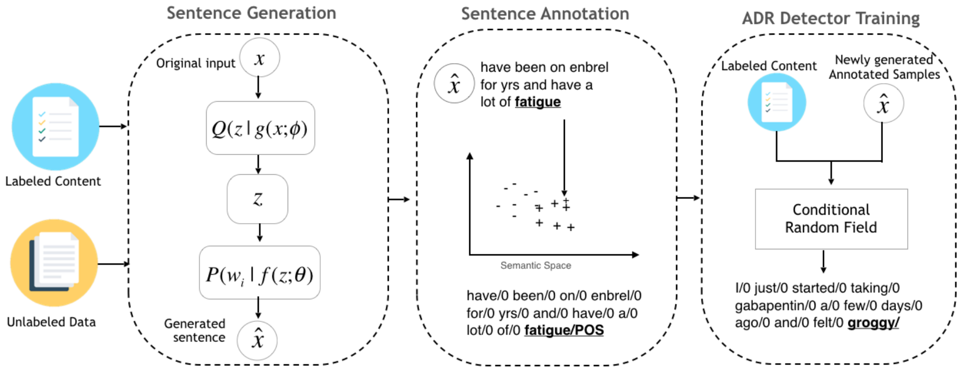
Sentence Generation
Sentence Annotation
ADR Detector Training
Original input
x
Labeled Content
$Q(z \mid g(x;\phi))$
z
Unlabeled Data
$P(w_i \mid f(z;\theta))$
Generated sentence
$\hat{x}$
have been on enbrel for yrs and have a lot of fatigue
Semantic Space
have/0 been/0 on/0 enbrel/0 for/0 yrs/0 and/0 have/0 a/0 lot/0 of/0 fatigue/POS
Labeled Content
Newly generated Annotated Samples
Conditional Random Field
I/0 just/0 started/0 taking/0 gabapentin/0 a/0 few/0 days/0 ago/0 and/0 felt/0 groggy/

Figure 1: Overview of the proposed Training Data Augmentation Approach for Adverse Drug Reaction Detection

3 Adverse Drug Reaction Detection in User Generated Content

Approach Overview. Figure 1 presents an overview of our proposed approach. Given a list of drug names, a corpus $UGC = \{ugc_1, .., ugc_n\}$ of health-related User Generated Content (UGC) that mention one or more drugs of interest, and a subset $LC \subset UGC$ of UGCs labeled with ADR mentions, the *Sentence Generation* step (Sections 3.1) creates a set SC of newly generated sentences that are similar to the ones in LC. The size of LC is usually highly limited, thus Sentence Generation is important to expand the labeled data for better training the ADR detector. In LC, terms related to ADR (e.g. "no appetite") are considered positive examples ($POSTerms$), while all the other terms (e.g. "aspirin", "again"), excluding English stop words, are considered negative examples ($NEGTerms$). A Variational Auto Encoder (VAE) is first trained on UGC data to learn the data distribution of the dataset, and then provided with LC sentences as input to generate the output set SC. The *Sentence Annotation* step (Section 3.2) then propagates the label from LC to the terms of sentences in SC. This is achieved by labelling the set $UTerms = \{ut_1, .., ut_n\}$ of terms in the newly generated sentence SC that are semantically more similar to $POSTerms$ than to $NEGTerms$ in LC. Finally, the labelled sets LC and SC are combined in the *ADR Detector Training* step (Section 3.3) to train an ADR detector.

3.1 Sentence Generation

Our method for data generation relies on learning sentence distributions from a large text corpus, which can then be used to generate posts SC semantically similar to a given set of existing labeled content LC. Let $\mathbf{x} \in \mathbb{R}^{|V|}$ ($\mathbf{x} \in UGC$) be the bag-of-words (multi-hot) representation of a user-generated content, where V is the global vocabulary, and $\mathbf{w}_i \in \mathbb{R}^{|V|}$ be the one-hot representation of the word at position i in the sentence represented by $\mathbf{x}$. Our goal is to learn $P(\hat{\mathbf{x}}|\mathbf{x})$, where the probability of a newly generated content $\hat{\mathbf{x}}$ serves as a proxy of the semantic similarity between $\hat{\mathbf{x}}$ and the original labeled content $\mathbf{x}$. Note that we will use the full set of user generated content UGC to learn the data distribution, while only the labeled subset LC is used to generate new sentences.

To obtain this conditional distribution, we adopt the deep generative modeling approach (Kingma and Welling, 2014; Miao et al., 2016), which was originally proposed to generate data instances similar to those already in a given dataset. Here, data is embedded into a latent space which is modelled by conditional distributions, and samples from this distributions can be decoded into new artificial data instances. In contrast to shallow models such as Skip-Gram (Mikolov et al., 2013) which also embeds into latent spaces, deep generative models have been shown to capture the implicit semantics and structure of the underlying data more effectively. However, existing deep generative models are not designed for generating class-specific data instances. Therefore, our goal is to extend existing deep generative models such that we can choose to only generate samples of a chosen subclass (e.g., resembling just labeled data). For example, Table 1 shows 3 artificial samples generated for 2 human-written training data sentences.

To do so, we build our method upon variational autoencoder (VAE), a representative deep generative model capable of learning high-quality representations of data structures. Given a set of sentences, VAE aims at learning a likelihood func-

Table 1: Three samples generated using VAE for a given input sentence.

Input	**my dr switched from celexa to paxil and paxil made me feel sick**
Sample 1	my doctor put me on cymbalta and cymbalta can help me function
Sample 2	took my fluoxetine and it was a bit spaced out of my brain
Sample 3	yeah have to take topamax and it helps me but still feel fuzzy headed to a bit
Input	***bruh this vyvanse putting me to sleep I needa take a break***
Sample 1	took my vyvanse today and my head is spinning
Sample 2	vyvanse makes me feel like a zombie
Sample 3	vyvanse and addy have a cup of coffee
Input	***I was on Prozac for months but it made my emotions so suppressed I stopped taking them***
Sample 1	I was on venlafaxine for anxiety and depression but it stopped working
Sample 2	I was on effexor for about 3 months and then switched to venlafaxine
Sample 3	was on latuda for a while but it didn't help me

tion $P_\theta(\hat{\mathbf{x}}|\mathbf{z})$ that, when used together with a standard Gaussian prior of $\mathbf{z}$, can generate new data instances $\hat{\mathbf{x}}$ that are similar to existing ones. Here $\mathbf{z}$ is the latent feature vector that captures the underlying data structure of the existing dataset. To handle the complex relationship between the latent feature and textual content, the likelihood function is parameterized by deep neural networks.

Variational Autoencoder. VAE encompasses a generative model, which describes the generative process for new data instances $\hat{\mathbf{x}}$ given $\mathbf{z}$ sampled from the Gaussian prior and transformed through a deep neural network.

- For each user-generated sentence $\mathbf{x}$
  - Draw a latent feature vector $\mathbf{z} \sim P(\mathbf{z})$ where $P(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$ is the standard Gaussian distribution.
  - For the i^{th} term in the sentence,
    * Draw $\mathbf{w}_i \sim P(\mathbf{w}_i|f(\mathbf{z};\theta))$

where $f(\mathbf{z};\theta)$ is the neural network whose weights are shared for all sentences. The conditional probability over words, i.e., $P(\mathbf{w}_i|f(\mathbf{z};\theta))$ is modeled by a multinomial logistic regression:

$$P(w_i|f(\mathbf{z};\theta)) = \frac{exp(\mathbf{w}_i^\intercal f(\mathbf{z};\theta))}{\sum_{j=1}^{|V|} exp(\mathbf{w}_j^\intercal f(\mathbf{z};\theta))}$$

The parameters of the neural network, i.e., θ, are learned by maximizing the the log likelihood of the observed sentence $\mathbf{x}$. This is non-trivial due to the intractability of the integral over the latent feature vector $\mathbf{z}$. VAE adopts a variational approach to optimise for the lower bound of the log-likelihood:

$$\mathcal{L} = \mathbb{E}_{Q(\mathbf{z}|g(\mathbf{x};\phi)}[\sum_{i=1}^{|\mathbf{x}|} \log P(\mathbf{w}_i|f(\mathbf{z};\theta)] \\ -D_{KL}[Q(\mathbf{z}|g(\mathbf{x};\phi)\|P(\mathbf{z})]$$

This is generally known as the evidence lower bound (ELBO) (Blei et al., 2017). In such an ELBO, $\mathbb{E}(\cdot)$ is the expectation and $D_{KL}[\cdot\|\cdot]$ is the KL-divergence between two distributions; $Q(\mathbf{z}|g(\mathbf{x};\phi))$ is a Gaussian distribution $\mathcal{N}(\mu, \text{diag}(\sigma^2))$ that is again parameterized by a deep neural network: the two parameters of the Gaussian distribution, i.e., μ and σ are both the output of the neural network $g(\mathbf{x};\phi)$.

New Content Generation. Once a VAE is trained on all user-generated content UGC, we take the existing human-annotated content LC (annotated with ADR mentions) as the input for VAE to generate new sentence SC. The generation is performed by making use of the two conditional distributions learned before, i.e., $Q(\mathbf{z}|g(\mathbf{x};\phi))$ and $P(\mathbf{w}_i|f(\mathbf{z};\theta))$. When used together, these distributions form the conditional distribution we are interested for generating new content:

$$P(\hat{\mathbf{x}}|\mathbf{x}) = \int \sum_{i=1}^{|\hat{\mathbf{x}}|} P(\mathbf{w}_i|f(\mathbf{z};\theta))\ Q(\mathbf{z}|g(\mathbf{x};\phi))d\mathbf{z}$$

Content generation can then be performed via sampling from the above distribution. To generate new sentences, we take each sentence from the labeled set LC, and sample a pre-defined number (k) of latent feature vectors $\mathbf{z}_{j=1}^k$ from $Q(\mathbf{z}|g(\mathbf{x};\phi))$. For each sampled $\mathbf{z}_j$, we use it as an input for $P(\mathbf{w}_i|f(\mathbf{z};\theta))$ to generate a sequence of words as the new sentence.

3.2 Sentence Annotation

After generating new samples SC similar to LC, the next step is to automatically annotate the terms in the newly generated sentences with ADR mentions such that it can be used to train a sequence-labeling model. In its basic version, we can only rely on the terms in the $POSTerms$ as positive examples of ADRs. However, we will heuristically expand this term set with additional positive examples found in the SC, thus improving the recall of the ADR detector.

In this work we rely on measuring and aggregating the semantic relatedness SR between a term ut_i and all the terms in $POSTerms$ as well as $NEGTerms$. In general, terms which are semantically related to terms in the $POSTerms$ should be considered as positive example. For example, having the terms `fever` and `no appetite` as positive examples, the new terms `weakness` or `body aches` could also be added to $POSTerms$ (because they are considered semantically related due to frequent co-occurrence, following the distributional hypothesis (Harris,

Table 2: CRF training parameters.

useNGrams=true	normalize=true
noMidNGrams=true	useOccurrencePatterns=true
usePrev=trueuseNext=true	useLastRealWord=true
useLemmas=true	useNextRealWord=true
maxLeft=1	lowercaseNGrams=true

1954)), while wheelchair shall be added to $NEGTerms$. To this end, we use the popular *word2vec* implementation of skip-n-gram word embeddings (Mikolov et al., 2013). We define the semantic relatedness $SR_{pos}(ut_i, POSTerms)$ for a term ut_i and the $POSTerms$ as well as $SR_{neg}(ut_i, NEGTerms)$ as follows:

$$\mathrm{SR}_{pos}(ut_i) = \frac{\sum_{pterm \in POSTerms} SR_{pos}(ut_i, pterm)}{|POSTerms|}$$

$$\mathrm{SR}_{neg}(ut_i) = \frac{\sum_{nterm \in NEGTerms} SR_{neg}(ut_i, nterm)}{|NEGTerms|}$$

Some terms are semantically related to both $POSTerms$ and $NEGTerms$; for instance, terms such as drugs, pills, and pharmacy have a very close SR_{pos} and SR_{neg}. In order to avoid noisy terms which have an overlap in positive and negative semantics, we only annotate a term as positive if it appears in the $POSTerms$; *or* if the semantic relatedness between ut_i and $POSTerms$ is higher than the semantic relatedness between ut_i and $NEGTerms$, and if the distance between SR_{pos} and SR_{neg} is higher than a given threshold *(th)*.

3.3 ADR Detector Training

The labeled training data generated in the previous step can then be used to train any kind of supervised sequence tagger for ADRs. Conditional Random Field (CRF) has shown to be an effective technique on different NER tasks (Lafferty et al., 2001).We used the popular Conditional Random Field (CRF) sequence model[1] trained using the features listed in Table 2. Finally, the trained ADR detector can be used to detect the ADR mentions in our desired user generated content.

4 Evaluation

4.1 Experimental Settings

We evaluate the performance of our approach using precision, recall and f-score via approximate matching (Tsai et al., 2006). The focus of our evaluation is on the variation of performance at different fractions of training data and the number of newly generated samples to demonstrate the effectiveness of our proposed approach in reducing costs of manual annotation for training.

Table 3: Dataset statistics. **LC**: labeled training set, **UGC**: unlabelled set. Number of sentences, words, and **unique** ADRs.

	LC(Training)			LC(Testing)			UGC	
Dataset	**Sentences**	**Words**	**ADRs**	**Sentences**	**Words**	**ADRs**	**Sentences**	**Words**
Twitter	693	6557	379	292	2601	154	146K	2.16M
Reddit	7506	133K	543	1820	31708	195	274K	3.65M

4.2 Datasets

Experiments are performed on two datasets targeting different Web platforms. We used the publicly available Twitter dataset from *PSB 2016 Social Media Shared Task* for ADR detection [2]. Next, to evaluate our approach on richer textual forum data, we manually created an annotated corpus of Reddit medical subreddits with the help of medical experts. The aforementioned datasets contain only labeled data, but our approach requires in addition a larger corpus of unlabeled data from the same source. We therefore expanded each datasets with new posts, crawled respectively from Twitter and Reddit, that contain at least of one of the drug names contained in a common vocabulary [3]. The properties of each dataset are described in Table 3.

Twitter. The *PSB 2016 Social Media Shared Task* Twitter dataset (i.e. collected as explained in (Nikfarjam et al., 2015)) is a widely used manually annotated training data for ADR detection. The original dataset contained a total of 2,000 tweet IDs[4]; at the time of this study we were able to retrieve text from only 643 tweets, which we acknowledge might have an effect on the performance of the trained models.

Reddit Data. Reddit is a discussion website where users share and discuss problems/ideas about different topics. Reddit also contains subreddits such as AskDocs[5], DiagnoseMe [6], or

[1] https://github.com/dat/stanford-ner. Details on the selected features: https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/ie/NERFeatureFactory.html.

[2] http://diego.asu.edu/psb2016/task2data.html

[3] http://diego.asu.edu/downloads/publications/ADRMine/drug_names.txt

[4] Due to Twitter's search APIs license, only tweet ids were released

[5] https://www.reddit.com/r/AskDocs/

[6] https://www.reddit.com/r/DiagnoseMe/

`Bipolar` [7] where users share information about their health-related issues. To create a labeled training data set, we used the set of drug names mentioned above to collect 1,626 Reddit posts containing at least one drug names. We then recruited a medical doctor to annotate the ADRs (mentions of adverse drug reactions) in the collected posts following the annotation guidelines suggested in (Karimi et al., 2015), which specify: 1) exclude Leading prepositions, qualifiers, or possessive adjectives from selecting the ADR span, to avoid inconsistency. For instance, in the sentence "`it increases my anxiety`" only `anxiety` should be annotated; and 2) annotate all relevant contexts for an ADR concept. For example, in the sentence "`I have a severe muscle pain`", "`severe muscle pain`" should be annotated (not just "`muscle pain`"). To validate the labels, two of the authors manually checked again the annotations and found some ADRs which were not detected by the annotator; also, ambiguous ADRs where identified and discussed with the medical expert. From all the annotated posts, 600 posts with 9,326 sentences contained at least one ADR which were split into training and testing as shown in Table 3.

4.3 Compared Methods

We compare our proposed approach to established state-of-the-art ADR detection algorithms of different types:

- **QuickUMLS (Soldaini and Goharian, 2016):** an approximate dictionary matching algorithm which relies on UMLS concepts. We used the following setting, mentioned in (Soldaini and Goharian, 2016) as having best performance: *Similarity threshold* $= 0.9$, *Semantic types* $= [SignorSymptom, DiseaseorSyndrome, Finding, NeoplasticProcess]$
- **CRF (Baseline)**. The Conditional Random Field Phrase Detection Model [8] trained on the manually annotated training data LC.
- **CRF+VAE (Proposed):** In our proposed approach, we train a CRF model on the expanded training data created using the Variational Auto-Encoder approach as discussed in Section 3.
- **BLSTM-RNN(Cocos et al., 2017):** A state-of-the-art Bidirectional Long Short Term Memory (BLSTM) recurrent neural network (RNN) trained on the manually annotated training data LC.
- **BLSTM-RNN+VAE (proposed):** We combined our proposed technique with the BLSTM-RNN phrase detection technique. This is to highlight that our method can be combined with any supervised phrase detection technique.

To demonstrate the effectiveness of different strategies for augmenting the training data for ADR phrase detection, we compare our proposed approach with the following techniques:

- **CRF+SelfTraining (Usami et al., 2011):** A simple semi-supervised learning technique, where we train a similar conditional-random field phrase detection model as described before, but we apply the trained model on a set of randomly selected unlabeled sentences from UGC (i.e. we used 500 samples). The sentences containing newly annotated ADRs are added to the initial training data and are used to re-train the phrase detection model.
- **CRF+Doc2vec:** CRF model trained on data expanded using an embedding-based strategy. Instead of generating new content SC using VAE, we use Doc2vec (Le and Mikolov, 2014) which is inspired by word2vec (Mikolov et al., 2013) to find sentences similar to the labeled content LC.

4.4 Training

For training the Variational Autoencoder described in Section 3.1, we set the word dropout to 0.5, the learning rate to 0.001 and we used GRU (Cho et al., 2014) for both the encoder and the decoder. For labeling the newly generated sentences, we used word embeddings as described in (Mikolov et al., 2013). For Twitter we used pre-trained word embeddings trained on Twitter as described in (Godin et al., 2015). Since these pre-trained word embeddings did not perform well on the Reddit dataset, we trained a custom word embedding on all our Reddit data. We trained the skip-gram *word2vec* (300 dimension) model on the whole Reddit unlabeled collection.

[7] https://www.reddit.com/r/bipolar/

[8] https://github.com/dat/stanford-ner

Table 4: Performance of the different ADR detection techniques on the Twitter and Reddit test sets.

Technique	Precision	Recall	Fscore
QuickUMLS	.47	.34	.39
CRF	.67	.42	.51
BLSTM-RNN	.61	**.87**	.72
CRF+VAE	.68	.49	.57
BLSTM-RNN+VAE	**.71**	.85	**.77**

(a) Twitter

Technique	Precision	Recall	Fscore
QuickUMLS	.14	.21	17
CRF	**.72**	.47	.57
BLSTM-RNN	.67	.28	.39
CRF+VAE	.69	**.52**	**.60**
BLSTM-RNN+VAE	.63	.29	.40

(b) Reddit

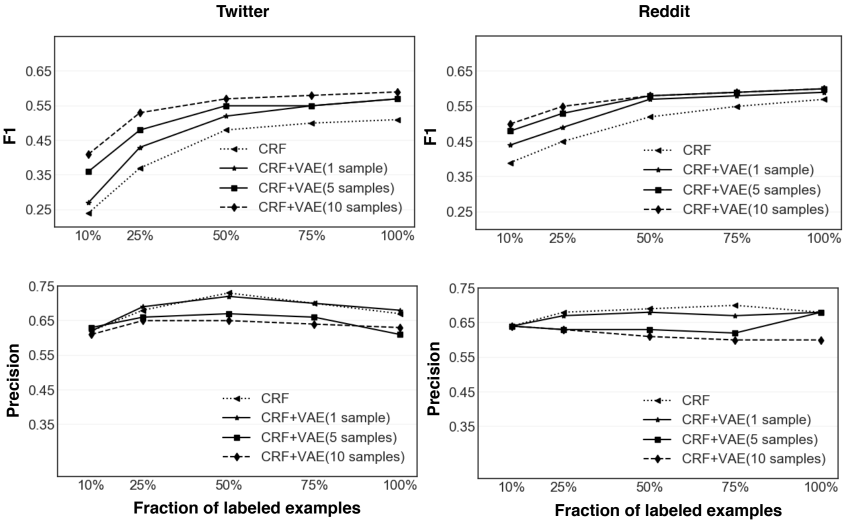

Figure 2: Average *F1* and *Precision* for *CRF* and *CRF+VAE* techniques, trained using different fractions of manually annotated examples and varying number of samples generated using VAE. Tested on the *Twitter* test set (on the left) and on the *Reddit* test set (on the right).

5 Results and Discussions

5.1 Comparison with ADR Detectors

In the first experiment, we compare our approach (i.e. trained with 100% of the labeled training data with 1 sample generated for each sample in the LC) against different ADR detector techniques described in Section 4.3. Table 4 reports precision, and recall and F1-measure, of all the baselines in comparison to proposed approach CRF+VAE in *Twitter* and *Reddit* dataset. We make the following observations: *QuickUMLS* is outperformed by all the other methods. The result shows that dictionary based approaches are not able to cover concepts that do not have a reference in UMLS dictionary, and produce false positives by labeling irrelevant words such as “maybe”, “energy”, “condition”, “illness”, or “worse” as positive.

The difference in performance between CRF and CRF+VAE shows the advantage brought by the sentence generation (VAE) and sentence annotation step of our approach. To highlight that our method can be combined with any supervised phrase detection technique, we combined our proposed technique with the BLSTM-RNN. BLSTM-RNN outperforms CRF in Twitter dataset; note that the model was designed to detect ADRs from the Twitter dataset. The results show that independent of the methodology used for training an ADR detector (e.g. CRF or BLSTM-RNN), expanding training data with VAE improves the overall performance. However due to the large amount of time required for training the BLSTM-RNN and the unstable prediction performance of its model on the test set (Cocos et al., 2017), the remaining experiments just focus on CRF for training ADR detector.

5.2 Effects of Training Data Size on CRF+VAE

For a given dataset (*Twitter* and *Reddit*), we created smaller subsets of the training data (i.e. 10%, 25%, 50%, 75%, 100%) to simulate the effect of limited training data availability. The subsets are randomly selected, and experiments are repeated 10 times for each size setting. We then train a CRF algorithm and different variants of our CRF+VAE (i.e. with different subsets of training data and different size of newly generated content for each labeled sample) and compare their performance. In particular, the core advantage of our approach is that we are able to generate any number of additional training data samples. Therefore, we test different settings where we generate an extra 1, 5, or 10 artificial sentences for each labeled sentence in the training set. The experiments are conducted 10 times for each setting.

Figure 2 summarizes the average performance achieved for *Twitter* and *Reddit* datasets. The results show that by using the VAE to expand the training data, it is possible to obtain higher F-scores for both datasets. In Addition, we can show that by increasing the number of artificially generated samples (i.e. 5 and 10 samples), we can achieve a considerable F-score boost up to (+.17) and (+.12) for Twitter and Reddit (i.e. with just 10% of the labeled samples). We did not observe any significant improvement with more than 10 samples. This limitation is likely due to our constraint to generate sentences similar to the existing annotated sentences instead of radically new ones - a limitation chosen to allow us to perform reliable label propagation which would be hard for sentences too different. The results also show that by generating 1 sample using VAE but only using 50% of the training data, we can obtain comparable results to using the 100% of the labeled training data without VAE. When generating more training samples (i.e. 10 samples), our approach can achieve comparable performance with only the 25% of the initial labeled set. As shown in Figure 2, the effect of VAE expansion is greater the smaller the training data set is, thus VAE is used most efficiently to reduce the training costs of ADR detection significantly while maintaining quality. Note that all the improvements of CRF+VAE over CRF are statistically significant using paired t-test (i.e. p-value<0.05). When artificially expanding training data, recall is often improved at the cost of precision. This is demonstrated by the performance of CRF+Doc2Vec (Table 5). However, even using CRF+VAE (1 sample) shows higher F-score than CRF without notable loss of precision. This positive behaviour can be attributed to the larger number of positive and negative examples which helps to maintain the generalisation capabilities of the ADR detector while refining the quality of its recognition.

5.3 Comparison of Data Expansion Techniques

In the third experiment, we compare the performance of CRF+VAE against the two other automatic training data expansion techniques CRF+SelfTraining and CRF+Doc2Vec. As in the previous experiment, we use 10%, 25%, 50%, 75% and 100% of the training data. For the sake of brevity, we only report the best performance[9] achieved by these techniques in Table 5.

CRF+SelfTraining keeps the precision high but compared to CRF+Doc2Vec and CRF+VAE, it is not able to increase the recall significantly. Its low recall can be attributed to treating some terms incorrectly as negative instance examples. This is due to relying only on the output of the trained model for labeling the training data for the next iteration. We observe that CRF+VAE achieves better precision and comparable recall to CRF+Doc2Vec with the Twitter dataset, while achieving similar performance in the Reddit dataset in terms of F-score, but with higher precision. This underlines that artificially generating new similar training sentences can outperform discovering existing similar training sentences using Doc2Vec similarity. The results show that our approach in general performs better in the Twitter dataset. This can likely be attributed to the differences in the structure between the two datasets. Each tweet contains on average 8 words, while each Reddit sentence contains on average 17 words. Also, VAEs have shown to perform better on shorter sentences (Semeniuta et al., 2017).

6 Qualitative Analysis

In this section we tested CRF+VAE approach on *Twitter* and *Reddit* test sets and manually inspect all the posts containing false positive and negatives to understand the reasons for the prediction errors.

False Positives. Manual inspection of the posts reveal that most of the false positives are due to 1) Mis-recognizing *indications* as an ADR, i.e. an illness for which the drug has been prescribed is recognized as an adverse drug reaction (Chowdhury et al., 2018). For instance in the two posts *"I started effexor after having pretty severe postpartum depression"* and *"depression hurts cymbalta can help"*, `depression` is labeled as ADR even though it is an *indication*. However,

[9] The Self-training configuration has been run for ten iterations; we report the iteration with best performance.

Table 5: Average Precision/Recall/F1 with standard deviation in parenthesis for CRF, CRF+SelfTraining, CRF+Doc2Vec and CRF+VAE on Twitter and Reddit datasets. The experiments are conducted 10 times for each setting.

Datasets	**%Labeled samples**	**CRF**	**CRF+SelfTraining**	**CRF+Doc2Vec**	**CRF+VAE**
	10	62(.1)/.15(.05)/.24(.07)	.65(.05)/.27(.05)/.38(.06)	.57(.09)/.30(.04)/.39(.04)	.61(.10)/.32(.04)/.41(.04)
	25	68(.06)/.25(.03)/.37(.04)	.66(.05)/.32(.02)/.43(.02)	.62(.03)/.42(.03)/.50(.03)	.65(.04)/.44(.03)/.53(.02)
Twitter	50	73(.02)/.35(.02)/.48(.02)	70(.03)/.37(.03)/.48(.03)	.65(.04)/.50(.01)/.56(.01)	.65(.01)/.51(.02)/.57(.01)
	75	70(.02)/.39(.01)/.50(.02)	68(.02)/.40(.02)/.51(.02)	.66(.02)/.52(.02)/.58(.01)	.67(.02)/.51(.03)/.58(.02)
	100	.67/.42/.51	.67/.41/.51	.61/.57/.59	.64/.56/.60
	10	.64(.06)/.28(.05)/.38(.05)	.64(.05)/.29(.05)/.40(.04)	.62(0.04)/.42(.04)/.50(0.3)	.64(.03)/.41(.04)/.50(.03)
	25	.68(.03)/.34(.03)/.45(.03)	.68(.03)/.34(.04)/.45(.03)	.61(.02)/.51(.02)/.55(.02)	.63(.02)/.48(.01)/.55(.01)
Reddit	50	.69(.02)/.42(.03)/.52(.02)	.69(.02)/.43(.04)/.53(.02)	.57(.02)/.60(.01)/.59(.01)	.61(.01)/.56(.02)/.59(.01)
	75	.70(.01)/.46(.02)/.55(.01)	.70(.01)/.46(.02)/.55(.01)	.56(.01)/.62(.01)/.59(.01)	.60(.01)/.59(.01)/.60(.01)
	100	.72/.47/.57	.71/.46/.57	.57/.64/.61	.60/.62/.61

`depression` commonly occur as ADR as well in other posts, which might be the cause for this error (Chowdhury et al., 2018); 2) Ignoring negative verbs. As an example the word `manic` in *"The only one that didn't make me manic, Wellbrutin"* and `vomiting` in *"@uclaibd I never had bleeding or vomiting just a lot of fatigue"* are detected as ADRs due to the structure of the posts. However the model was not able to distinguish the negative verbs; 3) Mis-labeling ADR-related words as an ADR: For instance in the post *"temperature would start to rise, depression weakens"* the word `depression` was recognized as ADR; 4) Mistakes in manual annotation in the test data. For instance in the Tweet *"Ive had no appetite since I started on prozac"*, the annotators did not annotate *no appetite* as an ADR. However, our model was able to predict it correctly as an ADR, but due to this mistake in test data is considered a false positive.

False Negatives. False negatives are likely to occur in posts that are ambigious or overly complex. For example, in the post *"Im just wondering if its safe to take tramadol 15h after vyanse and if promethazine and melatonin would lower my chances of a seizure"* the word `seizure` was not detected as an ADR. It must be noted how, in this specific case, even human annotators debated if seizure is indeed an ADR of tramadol, or an indication of vyanse. In another example *"Am I the only one that grinds the shit out of their teeth on Vyvanse"*. The expression `grinds the shit out of their teeth` is a long description of the slang ADR *teeth grind*, which has been described in a very unstructured and informal way. This is hard to handle for phrase detectors like CRF as some level of abstraction would be necessary to deal with this.

7 Conclusion

In this paper, we have demonstrated an approach for augmenting training data for detecting mentions of Adverse Drug Reactions from social media text in a very cost-efficient manner. We introduced a technique which expands human-labeled training sets with a large number of artificially generated training samples. The benefit of our training data generation technique is greater the smaller the manually created training data set is. Therefore, it is used most efficiently to reduce the manual training costs of ADR detection while maintaining quality (e.g., in our experiments, we can maintain quality even when reducing manually provided training data by 75%). Furthermore, we could show that our approach generally works better on Twitter data. We assume that this can be explained by Reddit forum posts using significantly richer, longer, and more complex sentences. VAEs are known to work more effectively with shorter sentences than with longer ones. This work is only one of the initial steps towards automated adverse drug effect analytics on social data. The next step would be to interpret the semantics of the extracted slang ADRs, and linking them to medical ontologies to allow for further structured analysis.

Acknowledgments

We thank Dr. Gerhard Mulder for his wonderful collaboration in annotating the Reddit dataset.

References

Syed Rizwanuddin Ahmad. 2003. Adverse drug event monitoring at the food and drug administration: your report can make a difference. *Journal of general internal medicine*, 18(1):57–60.

Eiji Aramaki, Sachiko Maskawa, and Mizuki Morita. 2011. Twitter catches the flu: detecting influenza epidemics using twitter. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1568–1576. Association for Computational Linguistics.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Brant W Chee, Richard Berlin, and Bruce Schatz. 2011. Predicting adverse drug events from personal health messages. In *AMIA Annual Symposium Proceedings*, volume 2011, page 217. American Medical Informatics Association.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *In Proceedings of the Empirical Methods in Natural Language Processing*.

Shaika Chowdhury, Chenwei Zhang, and Philip S Yu. 2018. Multi-task pharmacovigilance mining from social media posts. In *Proceedings of the 27th International Conference on World Wide Web*, pages 117–126. International World Wide Web Conferences Steering Committee.

Anne Cocos, Alexander G Fiks, and Aaron J Masino. 2017. Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts. *Journal of the American Medical Informatics Association*, 24(4):813–821.

Companion. 2019. Companion page. In *https://sites.google.com/view/emnlp-ijcnlp2019*.

Jan Deriu, Aurelien Lucchi, Valeria De Luca, Aliaksei Severyn, Simon Müller, Mark Cieliebak, Thomas Hofmann, and Martin Jaggi. 2017. Leveraging large amounts of weakly supervised data for multi-language sentiment classification. In *Proceedings of the 26th international conference on world wide web*, pages 1045–1052. International World Wide Web Conferences Steering Committee.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153.

Rave Harpaz, William DuMouchel, Nigam H Shah, David Madigan, Patrick Ryan, and Carol Friedman. 2012. Novel data-mining methodologies for adverse drug event discovery and analysis. *Clinical Pharmacology & Therapeutics*, 91(6):1010–1021.

Z. Harris. 1954. Distributional Structure. *Word*, 10:146–162.

Zhe He, Zhiwei Chen, Sanghee Oh, Jinghui Hou, and Jiang Bian. 2017. Enriching consumer health vocabulary through mining a social q&a site: A similarity-based approach. *Journal of biomedical informatics*, 69:75–85.

Trung Huynh, Yulan He, Alistair Willis, and Stefan Rueger. 2016. Adverse drug reaction classification with deep neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 877–887.

Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. 2015. Cadec: A corpus of adverse drug event annotations. *Journal of biomedical informatics*, 55:73–81.

Payam Karisani and Eugene Agichtein. 2018. Did you really just have a heart attack?: Towards robust detection of personal health mentions in social media. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 137–146. International World Wide Web Conferences Steering Committee.

Daniel Kershaw, Matthew Rowe, and Patrick Stacey. 2016. Towards modelling language innovation acceptance in online social networks. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 553–562. ACM.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. *stat*, 1050:1.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Int. Conf. on Machine Learning*, volume 951, pages 282–289.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Robert Leaman, Laura Wojtulewicz, Ryan Sullivan, Annie Skariah, Jian Yang, and Graciela Gonzalez. 2010. Towards internet-age pharmacovigilance: extracting adverse drug reactions from user posts to health-related social networks. In *Proceedings of the 2010 workshop on biomedical natural language processing*, pages 117–125. Association for Computational Linguistics.

Kathy Lee, Ashequl Qadir, Sadid A Hasan, Vivek Datla, Aaditya Prakash, Joey Liu, and Oladimeji Farri. 2017. Adverse drug event detection in tweets with semi-supervised convolutional neural networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 705–714. International World Wide Web Conferences Steering Committee.

Sepideh Mesbah, Christoph Lofi, Manuel Valle Torre, Alessandro Bozzon, and Geert-Jan Houben. 2018. Tse-ner: An iterative approach for long-tail entity extraction in scientific publications. In *International Semantic Web Conference*, pages 127–143. Springer.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International Conference on Machine Learning*, pages 1727–1736.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Azadeh Nikfarjam, Abeed Sarker, Karen Oconnor, Rachel Ginn, and Graciela Gonzalez. 2015. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, 22(3):671–681.

Abeed Sarker, Rachel Ginn, Azadeh Nikfarjam, Karen OConnor, Karen Smith, Swetha Jayaraman, Tejaswi Upadhaya, and Graciela Gonzalez. 2015. Utilizing social media data for pharmacovigilance: a review. *Journal of biomedical informatics*, 54:202–212.

Abeed Sarker and Graciela Gonzalez. 2015. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of biomedical informatics*, 53:196–207.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 627–637.

Luca Soldaini and Nazli Goharian. 2016. Quickumls: a fast, unsupervised approach for medical concept extraction. In *MedIR workshop, sigir*.

Chen-Tse Tsai, Gourab Kundu, and Dan Roth. 2013. Concept-based analysis of scientific literature. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1733–1738. ACM.

Richard Tzong-Han Tsai, Shih-Hung Wu, Wen-Chi Chou, Yu-Chun Lin, Ding He, Jieh Hsiang, Ting-Yi Sung, and Wen-Lian Hsu. 2006. Various criteria in the evaluation of biomedical named entity recognition. *BMC bioinformatics*, 7(1):92.

Yu Usami, Han-Cheol Cho, Naoaki Okazaki, and Jun'ichi Tsujii. 2011. Automatic acquisition of huge training data for bio-medical named entity recognition. In *Proceedings of BioNLP 2011 Workshop*, pages 65–73. Association for Computational Linguistics.

FM Zanzotto and Marco Pennacchiotti. 2012. Language evolution in social media: a preliminary study. *LINGUISTICA ZERO*.

Qing T Zeng and Tony Tse. 2006. Exploring and developing consumer health vocabularies. *Journal of the American Medical Informatics Association*, 13(1):24–29.